

Unclassified

Phase I Small Business Technology Transfer (STTR) Final Report
U.S. Government Contract Number: N00014-01-M-0220

Autonomous Distributed Systems

**- Multiple-Level Distributed Data Fusion For Future DADS Using Bayesian
Network Technology -**

15 January 2002

Period of Performance Reported: 1 July, 2001 to 15 January, 2002

Program Title: Autonomous Distributed Systems

Requisition/Purchase Request No.: 01PR10518-00/5JUN2001

Deliverable: CLIN 0001AD

Performing Organization Report Number: IET-TR ADS 04

Principal Investigator: Shozo Mori

Information Extraction & Transport (IET), Inc.

1911 N. Fort Myer Dr., Suite 600

Arlington, VA 22209

PH: (703) 841-3500; FAX: (703) 841-3501; EMAIL: smori@iet.com

University STTR Partner: Bruce D'Ambrosio

Oregon State University

Department of Computer Science

Dearborn Hall 303

Corvallis, Oregon 97331-3202

Consultant: Douglas M. Hart

Cyberneutics, Inc.

11654 Plaza America Dr.

Reston, VA 20190-4700

DISTRIBUTION STATEMENT A

**Approved for Public Release
Distribution Unlimited**

Sponsoring/Monitoring Agency: Office of Naval Research

Dr. Lawrence Green (Dr. David Johnson)

Ballston Tower One

800 North Quincy Street

Arlington, VA 22217-5660

Distribution Statement: Unlimited Distribution

Security Classification: Unclassified

Administrative POC: Jeffry Banes

Information Extraction & Transport (IET), Inc.

1911 N. Fort Myer Dr., Suite 600

Arlington, Virginia 22209

PH: (703) 841-3500; FAX: (703) 841-3501; EMAIL: jbanes@iet.com

20020322 165

Unclassified

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 15 January, 2002	3. REPORT TYPE AND DATES COVERED Final Report: 1 July, 2001 to 15 January, 2002		
4. TITLE AND SUBTITLE Autonomous Distributed System – Multiple-Level Distributed Data Fusion for Future DADS Using Bayesian Network Technology – Phase I STTR Final Report		5. FUNDING NUMBERS C: N00014-01-M-0220		
6. AUTHORS Shozo Mori, Information Extraction & Transport (IET), Inc. Bruce D'Ambrosio, Oregon State University Doug Hart, Cyberneutics, Inc.		8. PERFORMING ORGANIZATION REPORT NUMBER IET-TR ADS 04		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Information Extraction & Transport (IET), Inc. 1911 N. Fort Myer Dr., Suite 600 Arlington, VA 22209				
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research / Dr. Lawrence Green (Dr. David Johnson) Ballston Tower One 800 North Quincy Street Arlington, VA 22217-5660		10. SPONSORING / MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unlimited Distribution		12b. DISTRIBUTION CODE		
ABSTRACT (Maximum 200 words)				
<p>This final report summarizes technical achievements made for Phase I STTR project, entitled Autonomous Distributed System – Multiple-Level Distributed Data Fusion for Future DADS Using Bayesian Network Technology -, during the period from 1 July 2001 to 15 January 2002. The future DADS, also known as mini-DADS or micro-DADS, is the next-generation DADS envisioned for 2020 littoral undersea threats. The most prominent feature of the future DADS is the elimination of a fiber line for an acoustic arrays for each of the current DADS sensor node, which poses a wide variety of technical challenges to implement distributed data fusion, including target detection, tracking, and classification. The Phase I efforts are to show technical feasibility of application of the distributed Bayesian network inference algorithms developed by IET and OSU in the past decade. Communication between nodes is conducted using Bayesian networks fragments to represent local information, and Bandwidth Agile Situation Dissemination technique for transmitting Bayesian network fragments in face of communication limitations.</p>				
14. SUBJECT TERMS Mission Area: Intelligence, Surveillance, and Reconnaissance Deployable Autonomous Distributed System, Distributed Data Fusion Bayesian Networks, Influence diagram		15. NUMBER OF PAGES 85		
		16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT Unlimited Distribution	

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-1
298-102

Executive Summary

Information Extraction & Transport, Inc. (IET) and Oregon State University (OSU) are pleased to submit the final report for the Phase I Small Business Technology Transfer (STTR) project, Contract N00014-01-M-0220, entitled Autonomous Distributed System, to summarize technical progress made during the period from 1 July 2001 to 15 January, 2002. This final report is presented in fulfillment of contract line item CLIN/SLIN 0001AD. The objective of this STTR project is to develop new, innovative distributed data fusion concepts for networks of distributed autonomous sensors applying the Bayesian Network technology built through the last decade of collaboration between IET and OSU. The specific Navy application is to develop new distributed data fusion concepts for the *Deployable Autonomous Distributed System (DADS)* in order to fuse information from a field of autonomous sensors for surveillance purposes in littoral waters.

The DADS program is currently in its 6.3 stage and several field tests are in progress. A DADS sensor node consists of an acoustic array, a magnetic sensor, and acoustic modem to communicate with other DADS nodes. This deployable sensor node is battery operated and totally autonomous. However, it has become apparent that the long fiber line connecting the array of acoustic elements in a DADS sensor node makes it vulnerable to trawling and dredging activities. In order to avoid this vulnerability, a future DADS concept is being formed in which each single acoustic sensor element becomes an autonomous sensor node to eliminate the fiber lines connecting acoustic elements. Thus the most basic formation of a future DADS sensor node consists of a single acoustic element, with a magnetic sensor, and possibly with other "future" sensors.

This future DADS concept generates several technical challenges. The immediate challenge is to realize acoustic array functionality by distributed acoustic elements connected only by wireless communication (most likely acoustically, but possibly by other means in the water). The future DADS concepts also pose a unique set of challenges for distributed data fusion and communication, because of the increased needs for collaboration among sensor nodes. Furthermore, the environments that the possible future DADS would face may be quite different, anticipating an extremely wide variety of targets including: quiet (maybe smaller) submarines, surface combatants and vehicles, mines and mine deployment

platforms, swimmer delivery vehicles, Autonomous Underwater Vehicles (AUVs), Unmanned Underwater Vehicles (UUVs), high speed torpedoes, groups of small units, *etc.* This means that target classification tasks would become more than an order-of-magnitude more difficult, in addition to difficulties in detection and tracking such targets.

The IET/OSU team proposes to respond to these challenges with new, innovative distributed data fusion concepts that apply Bayesian Network technology. A Bayesian Network is mathematically defined as a network of random variables connected by arcs that represent probabilistic causality, *i.e.*, conditional probabilities, and is a natural extension of Markov chains, so that any complex probabilistic relationship can be precisely formulated. Any target and sensor model can be represented by a Bayesian Network, and once an observation or an extracted feature value is available as a realization or instantiation of random variables, target states, including position, velocity, target types, operational mode, *etc.*, can be *estimated* or *inferred*, using any of several Bayesian Network inference algorithms. IET/OSU has developed one of the most effective Bayesian Network inference algorithms, called the *Symbolic Probabilistic Inferencing* (SPI) algorithm. Recently, two new concepts, *Bayesian Network Fragments* and *Bandwidth Agile Situation Dissemination*, were also developed by the IET/OSU team, to be used for communicating aggregated information among distributed processing nodes.

The Phase I STTR project that this report describes is to develop distributed data fusion concepts for the future DADS by applying these two newly developed concepts. The Phase I efforts have been devoted formulating new distributed data fusion concepts, as described in this reports, to serve as a foundation for the proposed Phase II efforts. In the proposed Phase II efforts, these new concepts will be realized in terms of new algorithms, performance evaluation and prediction, and software development plans, to be used in the future DADS concept.

Table of Contents

1	Introduction	1
2	Target and Sensor Models	6
2.1	Target Model	6
2.2	Sensor Model	7
2.2.1	Acoustic Sensor Elements	8
2.2.2	Magnetic Sensor Elements	9
2.3	Processing Node Model	10
3	Data Fusion Distribution Levels	14
3.1	Intra-Sensor Data Fusion	14
3.2	Inter-Sensor, Intra Cluster Data Fusion	16
3.3	Inter-Cluster, System-Wise Data Fusion	17
4	Distributed Data Fusion Solutions	19
4.1	Target Detection	19
4.1.1	Single-Sensor Acoustic Detection	19
4.1.2	Multiple-Sensor Acoustic Detection	20
4.1.3	Single-Sensor Magnetic Detection	21
4.1.4	Multiple-Sensor Magnetic Detection	22
4.1.5	Multiple-Sensor Distributed Detection	24
4.2	Target Localization and Tracking	25
4.2.1	Acoustic TDOA Localization and Tracking	25
4.2.2	Acoustic Doppler Tracking	29
4.2.3	Magnetic Localization and Tracking	31
4.2.4	Inter-Cluster Distributed Tracking	34
4.3	Dynamic, Distributed, Hierarchical Classification Using Bayesian Networks	35
4.3.1	Problem Statement	35
4.3.2	Local Classification	37
4.3.2.1	Structuring Classification Knowledge for Representation and Application	37
4.3.2.2	Dynamic Classification	40
4.3.2.3	Multiple Targets	41
4.3.3	Distributed Classification	41
4.3.3.1	Communication Contents	41
4.3.3.1.1	Experiment Results	46
4.3.3.1.2	Discussion	49
4.3.3.2	Communication Timing	50
4.3.3.3	Communication Method	50
5	Conclusions	52
	References	54
	Appendix A: Bayesian Networks	58
	Appendix B: Generalized Likelihood Test: Neyman-Pearson Criteria	60
	Appendix C: Bandwidth Agile Situation Dissemination	67

Figures

Figure 1: DADS Sensor Node Concept	2
Figure 2: Future DADS Concept: Multiple-Level Distributed Data Fusion	4
Figure 3: Target States and Observations (Evidence)	7
Figure 4: Information Stored at Sensor Node	10
Figure 5: Distributed Data Fusion Using Bayesian Network	12
Figure 6: Micro DADS Deployment Patterns (Field Control)	15
Figure 7: Data Fusion in Transit	18
Figure 8: Acoustic Detection Probability	20
Figure 9: Multiple-Sensor Acoustic Detection	21
Figure 10: Target Localization by Single Magnetic Detection	22
Figure 11: Target Localization by Two Simultaneous Magnetic Detections	23
Figure 12: Target Localization by Positive and Negative Information from Two Magnetic Sensors	23
Figure 13: Hyperbolic Surface Determined by TDOA	26
Figure 14: TDOA Tracking	26
Figure 15: Hyperbola Intersection Method	27
Figure 16: Likelihood Function of Hyperbola Intersection Localization Algorithm	28
Figure 17: Doppler Frequency Measurements	29
Figure 18: Received Sound Pressure Measurements	30
Figure 19: Multiple-Sensor Doppler Tracking Data Fusion	30
Figure 20: Two-Sensor Magnetic Localization (Vertical)	32
Figure 21: Two-Sensor Magnetic Localization (Horizontal)	33
Figure 22: Hierarchical Representation of Classification Knowledge	38
Figure 23: Simplified Typical Platform Model	39
Figure 24: Sample Classification Tree for an Unknown Platform	39
Figure 25: Target-Sensor Model by Bayesian Network	42
Figure 26: Magnetic Sensor Node Classification Model with 2 nd Order Acoustic Likelihood Fragment Attached	44
Figure 27: 0-th Order (No Parent) Structural Model of Likelihood on State	45
Figure 28: 1-st Order (1 Parent Max) Structural Model of Likelihood on State	46
Figure 29: 2-nd Order (2 Parent Max) Structural Model of Likelihood on State	46
Figure 30: 3-rd Order (3 Parent Max) Structural Model of Likelihood on State	46
Figure 31: KL Distance from True Posterior (Vertical) by Fragment Size (Horizontal)	47
Figure 32: KL Distance from True Posterior, with 0-th Order Distance Normalized to 1	48
Figure 33: Log KL Distance from True Posterior, with 0-th Order Distance Normalized to 1	49
Figure 34: Overall Communication Functional Architecture	51

Tables

Table 1: DADS Distributed Data Fusion Levels for Future DADS	5
--	---

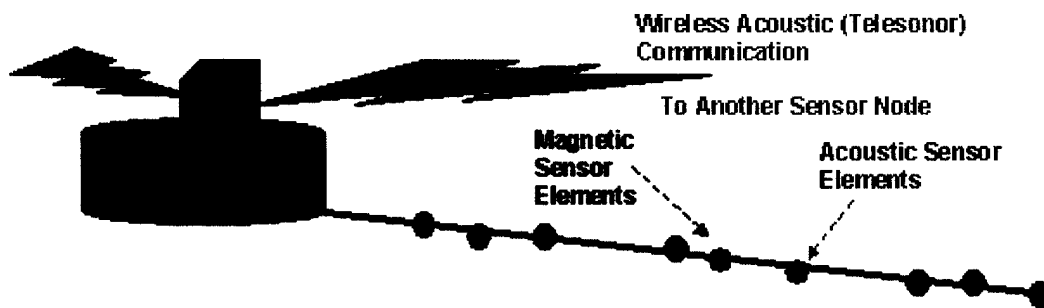
1 INTRODUCTION

A kickoff meeting for this Small Business Technology Transfer (STTR) program, as a collaboration of Information Extraction & Transport (IET) and Oregon State University (OSU), was scheduled for 26 September, 2001, with the ONR technical and management team, headed by Dr. David H. Johnson, at ONR, Arlington, VA. Prior to this meeting, IET and OSU had a meeting with the Deployable Autonomous Distributed System (DADS) Data Fusion Group, lead by Ms. Joan Kaina, at SPAWAR Systems Center, San Diego, in August, 2001, to be briefed on the subjects of the DADS data fusion programs and the future DADS concepts. Through these meetings, we reached a conclusion that, for our Phase I and possible Phase II STTR efforts, our technical focus should be *Multi-Level, Distributed Data Fusion for Future DADS using Bayesian Network Technology*.

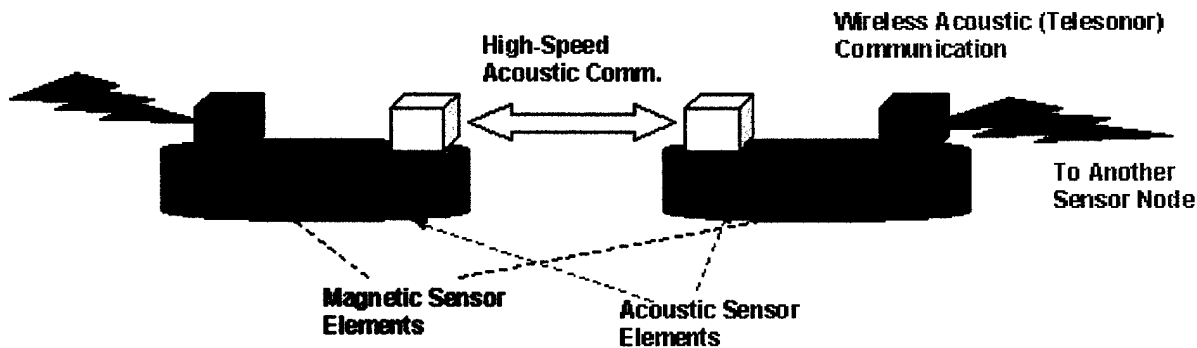
By the *future DADS*, we mean the next-generation DADS concept, commonly referred to as *mini DADS* or *micro DADS*, envisioned to be developed in a span ending in 2020 to 2030. Figure 1 contrasts the current and the future DADS sensor node concepts. One objective of the future DADS is to prepare for the 2020-2030 littoral threats, including quieter and smaller submarines, surface combatants and vehicles, mines and mine deployment platforms, swimmer delivery vehicles, Autonomous Underwater Vehicles (AUVs), Unmanned Underwater Vehicles (UUVs), and high speed torpedoes. Unlike the present threats, the future threats are in general groups of small units.

In order to counter these threats by generating finer and more sensitive fields of autonomous sensors, the future DADS concept uses a single sensor element for each sensor node ([8]), as shown in Figure 1b, as opposed to a current DADS sensor node that is equipped with a linear array of acoustic and magnetic sensor elements ([1] – [7]), as illustrated in Figure 1a. One of the reasons for the new DADS concept is to eliminate the vulnerability caused by each sensor node having a long line connecting sensor elements (vulnerability to dredging, trawling finishing, etc.).

This single-element sensor node concept generates a host of technical challenges, mainly because of limitation on communication that limits coherent processing capability, resulting in loss of observability dimensions. On the other hand, this new concept opens up possibility of a wide variety of new technology to be applied. Although a wide range of different sensors, including optical, seismic, chemical sensors, *etc.*, is being considered for the future DADS concept [8], the ONR's direction is for us to concentrate on sensor elements that we understand reasonably well, *i.e.*, acoustic and magnetic sensors.



(a) Current DADS Sensor Node Concept



(b) Future DADS (mini-DADS or micro-DADS) Sensor Node Concept

Figure 1: DADS Sensor Node Concept

Figure 2 illustrates the concept of the multiple-level distributed data fusion. Three levels of distributed data fusion are:

Unclassified

- (1) *Intra-Sensor-Node Level:* This is the data fusion inside each sensor element consisting of an acoustic and a magnetic sensor elements. Data fusion is between two sensor elements and for time-series of accumulated data.
- (2) *Inter-Sensor-Node, Intra-Cluster Level:* A small number (up to 12) of sensor nodes will form a sensor cluster. High speed communication among sensor nodes within a cluster may make it possible to perform coherent acoustic processing from which an instantaneous target localization may be possible. Two or more magnetic sensor elements within a cluster may provide short-range but good instantaneous three-dimensional target localization.
- (3) *Inter-Cluster, System-Wide Level:* Sensor nodes will be connected through wireless acoustic communication to each other, and through relays, finally to a surface gateway. At this level, cooperation among clusters is coordinated and the best picture will be provided to the external higher-level nodes, afloat and shore, through an RF gateway.

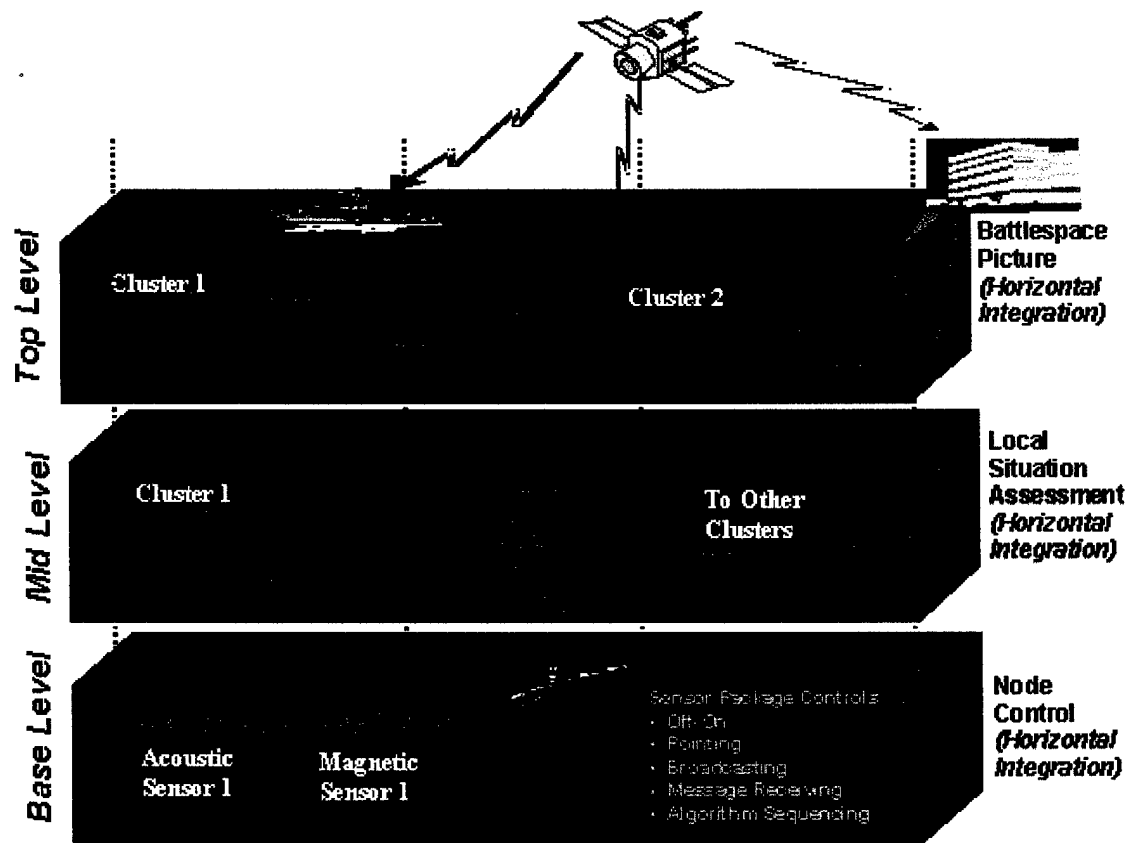


Figure 2: Future DADS Concept: Multiple-Level Distributed Data Fusion

We should not confuse these distributed data fusion levels with the JDL (Joint Directorate of Laboratories) defined Data Fusion Levels ([28]). As far as these JDL levels are concerned, we are directed by the ONR to concentrate on Level 1, *i.e.*, target detection, target localization and tracking, and target classification. We are also directed not to put our focus on data association aspects at least for the time being.

Table 1 shows the technical discussions for various future DADS distributed data fusion levels. In this table, the section for each topic in this report is shown by its section number. As seen in this table, this report does not address all of the topics reflecting the topics that are yet to be covered in the proposed Phase II efforts.

The approach taken by the IET/OSU team is to apply the *Bayesian Network technology* to these various levels of data fusion. The Bayesian Network techniques are probabilistic inference and estimation methods using a network, *i.e.*, a directed graph, of random variables and vectors, as shown in the short overview in Appendix A. Using graphic structures, Bayesian Networks

provide us with powerful yet rather intuitive methods for modeling complex probabilistic systems, and particularly, enable us to express causal relationships explicitly. IET and Oregon State University have developed a unique and powerful set of algorithms, called the *Symbolic Probabilistic Inferencing (SPI)* algorithms that effectively solves inference and estimation problems formulated using Bayesian Networks.

Table 1: Distributed Data Fusion Levels for Future DADS

Future DADS Distributed Data Fusion Level	Acoustic Sensor	Magnetic Sensor	Acoustic-Magnetic Fusion
Level 1: Intra-Sensor Node Data Fusion (3.1)	Detection Range (Range/Source SP) (4.1.1.) Doppler Frequency Tracking (Temporal Integration) (4.2.2) Target Classification Based on Acoustic Signature (4.3)	Detection Range (4.1.3) Estimation on Manifold in Source Magnet / Target Location Space Temporal Integration	Acoustic/Magnetic Detection Fusion (4.1.5) Acoustic/Magnetic Temporal Integration
Level 2: Inter-Sensor Node Data Fusion (3.2)	Multiple-Sensor Acoustic Detection (4.1.2) TDOA measurements by Coherent Processing (4.2.1) Multiple TDOA Localization (4.2.1) Multi-Sensor Doppler Tracking (4.2.2) Multi-Sensor Target Classification Fusion (4.3)	Multiple-Sensor Magnetic Detection (4.1.4) Multiple-Sensor Magnetic Localization (4.2.3) Magnetic Target Classification (4.3)	Multiple-Sensor Detection Fusion (4.1.5) Acoustic/Magnetic Localization Fusion Acoustic/Magnetic Classification Fusion
Level 3: Inter-Node System-Wide Data Fusion (3.3)	Multiple-Sensor Distributed Tracking and Classification (4.2.4) Information Relay to Surface RF Gateway Data Fusion in Transit: Alarming, Warning, and Hand-Over System-Wide Resource Management: Power Consumption Management and Control, Threshold Control		

The solution techniques developed through the Phase I efforts are described in Sections 2 to 4, as outlined in Table 1. The last section, Section 5, will describe conclusions and recommendations towards the further development of the Bayesian Network based data fusion concepts for the future DADS.

2 TARGET AND SENSOR MODELS

In this section, we will discuss target and sensor models. The level of the details for these models is intentionally kept very simple but with enough mathematical essence for the purpose of the solution concept development for our Phase I efforts. We will also briefly discuss our basic approach to the distributed data fusion problems, *i.e.*, the distributed Bayesian Network inference method. Its details will be described subsequently section 4.3 using a hierarchical target classification as an example.

2.1 Target Model

A target is modeled by its state. Let $x(t) = (\text{VEL}(t), \text{POS}(t), \text{ACS}(t), \text{MAG}(t), \text{TYPE})$ be the state of a target, at time t . $\text{VEL}(t)$ is the three-dimensional velocity vector and $\text{POS}(t)$ is the three-dimensional position of the target at time t , so that $(\text{VEL}(t), \text{POS}(t))$ is the target's six-dimensional geolocational state. $\text{ACS}(t)$ is the acoustic state and $\text{MAG}(t)$ is the magnetic state. In a simplest model, the $\text{ACS}(t)$ is the source-level sound pressure and the base frequency. In reality, the target's acoustic state $\text{ACS}(t)$ contains a potentially rich source of information such as various kinds of acoustic signatures, on which a variety of target classification algorithms may be performed to inference on the target type, TYPE , which is time-invariant. Detailed discussions on that aspect will be found in Section 4.3. The $\text{MAG}(t)$ is the three dimensional magnetic moments (as approximated by a point source). $\text{MAG}(t)$ may be considered time invariant.

The *a priori* causal relationship among those state components can be depicted by a Bayesian network representation (see Appendix A for a very brief introduction to Bayesian Networks) in Figure 3a.

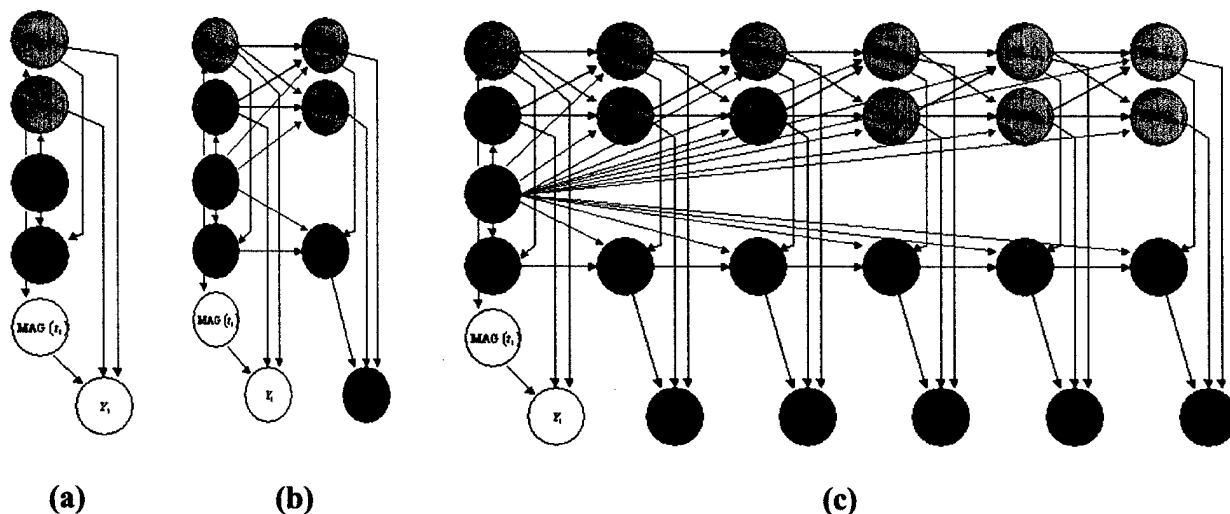


Figure 3: Target States and Observations (Evidence)

In Figure 3a, measurements from a magnetic sensor are represented by a random vector Y_1 that is called an *evidence node* in this Bayesian network (See Appendix A).

Figure 3b shows a probabilistic model that explains magnetic sensor measurements Y_1 obtained at time t_1 and acoustic measurements Y_2 obtained at time t_2 , simultaneously. Assuming that a track is initiated by a magnetic measurement vector and then continued only by acoustic measurements, Figure 3c illustrates a sequence of measurements and their relationship to the target states at different times.

The arcs connecting target state nodes in Figure 3 across different time samples represent the target model expressed by a set of conditional probabilities, *i.e.*, state transition probabilities. For example, the kinematic state component $(VEL(t), POS(t))$ can be any of the maneuvering target model, or the Constant Course and Speed (CCS) model for any length of time interval, or with a slight modification, can be modeled as multiple model dynamics.

2.2 Sensor Models

The sensor is modeled by the target-state-to-measurement transition probability, which is a conditional probability of a measurement vector given a target state vector, which is attached to each of the arcs connecting target state nodes and measurement nodes in Figure 3. Like the

target model, sensor models described in this section are only for the purpose of the discussions described in the later sections of this report, and their validity may be questioned in the further studies, as necessary.

2.2.1 Acoustic Sensor Elements

A simplified model for an acoustic sensor consists of (i) acoustic sound pressure measurement model (wide-band processing), (ii) base frequency measurement model (narrow-band processing), and (iii) acoustic spectrum pattern (acoustic signature) models (narrow-band processing). A simple sound pressure model would be

$$y_s = \frac{s}{r^2} + n_s \quad (1)$$

under the hypothesis H_1 that assumes that a target exists and under the condition that the target's source-level sound pressure is s and the target range from the sensor is r , where n_s is an independent zero-mean random variable with a unit that normalizes the sensor gain. Under the null hypothesis H_0 assuming the measurement is in the noise, we have the model $y_s = n_s$.

Under H_1 , the base frequency observation is modeled as

$$y_f = \left(1 - \frac{\dot{r}}{c}\right) f_s + n_f \quad (2)$$

where \dot{r} is the range rate of the target relative to the sensor location, c is the speed of sound in the medium (water), f_s is the source (base) frequency, and n_f is an independent zero-mean random variable.

The third observation may be an acoustic signature that can be used as the basis for the target classification. Theoretically, any single acoustic sensor element that is autonomously operated with sufficient power to support a necessary signal processing unit can produce a probability distribution, $\text{Prob}\{\text{TYPE} = T|Y\}$, for each possible target type T and any set of data Y accumulated to each moment. Although the exact functional form of this *a posteriori* probability distribution is not known, since this function can be autonomously performed for

each sensor element, target classification can be processed all through the three levels of distributed data fusion. This aspect will be discussed later in Section 4.3.

In a current DADS sensor node, a linear array of acoustic sensor elements is used to generate bearing measurements with coherent signal processing (to produce Time Difference Of Arrival (TDOA) observations) and with time series of such data through a Track Before Detect (TBD) tracking technique [1]. For the mini or micro DADS, however, such coherent acoustic processing must be done using high-speed, wireless acoustic communication. Whether such communication is possible with enough communication speed is still in question, depending on many factors, such as power consumption, sensor element spacing, modulation technology, and environmental parameters. Assuming this coherent processing is possible, target localization through acoustic elements is only possible at the second level of distributed data fusion, *i.e.*, the inter-sensor, intra cluster processing. This aspect will be discussed in Section 4.2.1.

2.2.2 Magnetic Sensor Elements

According to [1], a magnetic sensor element measures a three-dimensional magnetic field at the sensor location as, with appropriate unit normalization,

$$y_m = G(\Delta u)m + n_m \quad (3)$$

where $\Delta u = u - u_s$ is the three dimensional relative target-sensor positional vector with the three-dimensional target position u and the three-dimensional sensor location u_s , $G(\cdot)$ is the 3×3 matrix valued function defined as

$$G(\Delta u) = \|\Delta u\|^{-3} \left(3 \left(\frac{\Delta u}{\|\Delta u\|} \right) \left(\frac{\Delta u}{\|\Delta u\|} \right)^T - I \right) \quad (4)^1$$

m is the three-dimensional source magnetic moment vector, and n_m is an appropriate random vector that model background noises. In other words, under the hypothesis H_1 assuming that

¹ By X^T , we mean the transpose of a matrix or vector X . $\|x\|$ is the usual Euclidean norm of a vector x in any Euclidean space, *i.e.*, $\|x\| = \sqrt{x^T x}$. I is the identity matrix with an appropriate dimension.

there is a target at location u , the conditional probability of the three-dimensional magnetic field measurement y_m is determined by eqn. (3).

Depending on the dimension of the unknown components of the source magnetic moment m , even when the measurement error in eqn. (3) is negligible, we cannot determine the three-dimensional target location u . However, the single sensor element may produce target position-velocity estimates through the time-series, in particular, from a Closest Point of Approach (CPA) event. Alternatively, even with two sensor elements, simultaneous observation of the two three-dimensional magnetic field observations at different locations, all together six-dimensional observation, may produce target location and source magnetic moment estimates simultaneously. In other words, magnetic sensor elements may produce target localization estimates through the second level of distributed data fusion, *i.e.*, inter-sensor, intra-cluster data fusion.

2.3 Processing Node Model

As information is accumulated into each sensor node (that is itself an autonomous sensor node in the micro DADS concept), the target state estimation will be refined and each sensor node maintains its own target state estimates. At a certain point such information will be exchanged among sensor nodes, first within a sensor cluster, and then spread through the entire system.

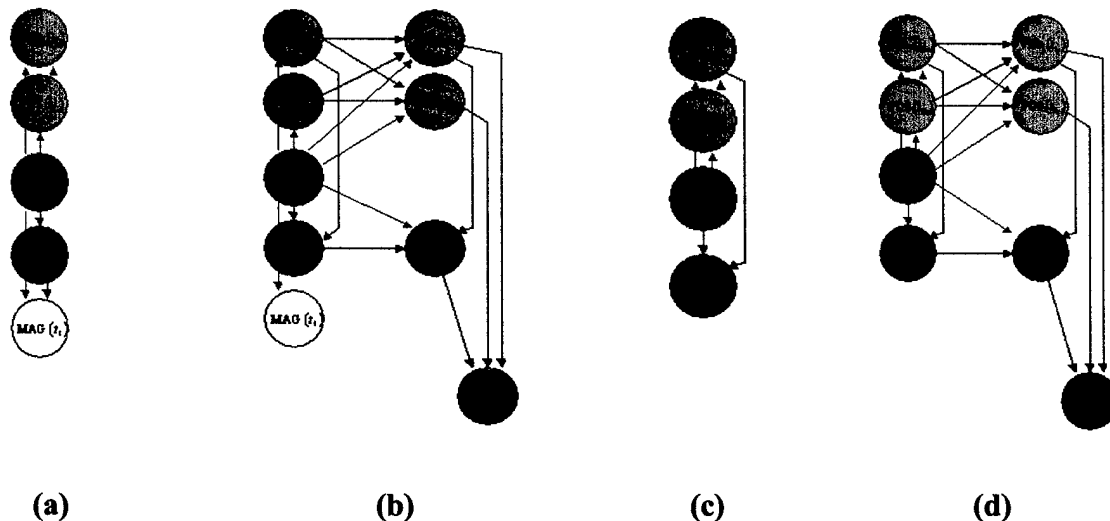
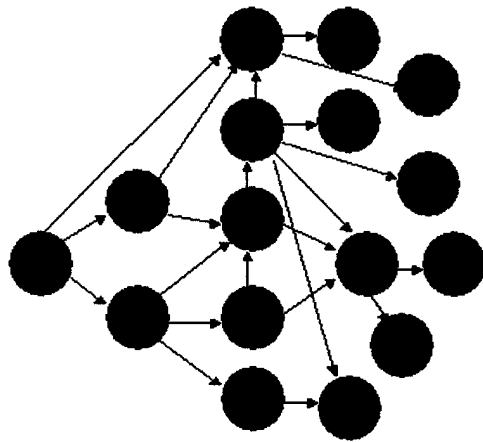


Figure 4: Information Stored at Sensor Node

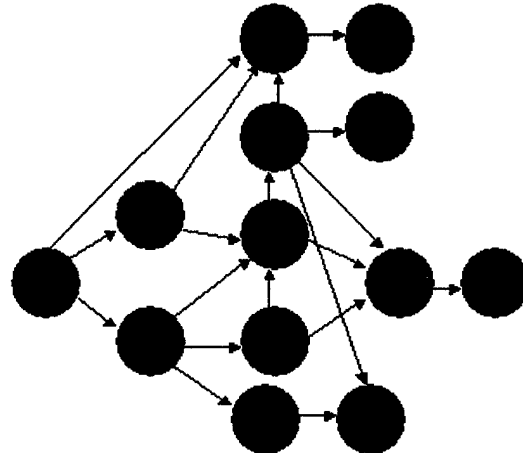
At each point, the information possessed by a sensor node can be represented by an *a posteriori* joint probability distribution, which can be represented by a Bayesian network, or more precisely, a Bayesian network fragment [31].

Figure 4 illustrates the changes in the informational states in each sensor node. Figure 4a illustrates an initial informational state, *i.e.*, the *a posteriori* probability distribution that is obtained by updating the *a priori* probability distribution by the first measurement that is assumed to be a magnetic sensor measurement. Figure 4b illustrates the procedure in which the *a posteriori* distribution shown in Figure 4a becomes an *a priori* probability distribution with respect to the new acoustic measurement Y_2 through an extrapolation procedure. Figure 4c is the Bayesian network fragment that represents the *a posteriori* probability constructed from the first two measurements, Y_1 and Y_2 . Then Figure 4d illustrates the updating procedure by yet new measurement Y_3 .

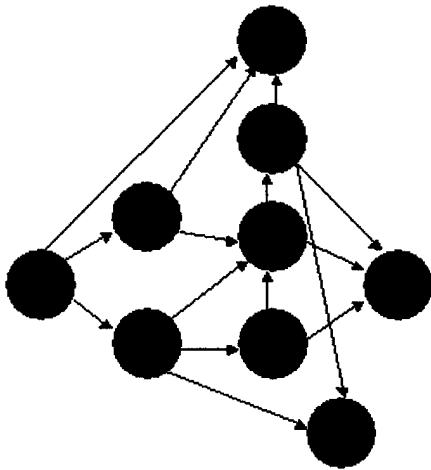
Figure 5 illustrates the distributed data fusion solution concept using the Bayesian network inference method. Figure 5a is a Bayesian network presentation of the two sensor data fusion problems. For the purpose of illustration, in Figure 5, the target states are assumed to consist of: (1) the target type (TYPE), (2) the operation mode (MODE), (3) the mass of the target (MASS), (4) the target position (POS), (5) the target velocity (VEL), (6) the engine output (Eout), (7) the primary acoustic signature (AS1), (8) the secondary acoustic signature (AS2), and (9) the target source vector magnetic moment (MS). Given those states, the two sensors, or two sensor clusters, receive observations consisting of (i) the primary acoustic signature measurements, AS11 and AS12, by Sensors 1 and 2, (ii) the secondary acoustic signature measurements, AS21 and AS22, and (iii) the vector magnetic field measurements, MM1, by Sensor 1, and (iv) the positional measurements, POS1 and POS2, by two sensors. In reality, as described in Section 3, the positional information is propagated from the target state to the sensor measurements in a much more complicated way. However, we use a simplified picture in Figure 5 for illustrative purposes.



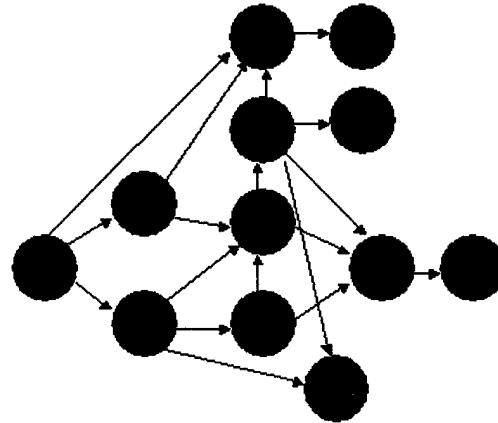
(a) Centralized Solution



(b) Local Solution 1



(c) Bayesian Network Fragment



(d) Fusion of Solutions

Figure 5: Distributed Data Fusion Using Bayesian Network

Figure 5a illustrates the centralized data fusion problem in which the two sensors provide measured information to a hypothetical central data fusion node. Since each sensor is operated autonomously, when Sensor 1 detects a target as both acoustic and magnetic detections, Sensor 1 evaluates the Bayesian network that models local information processing by Sensor 1 as shown in Figure 5b. Sensor 1 then tries to send information to Sensor 2, and when it receives that information, Sensor 2 will fuse information sent by Sensor 1 with its own local information. To do this, the target states updated by Sensor 1 measurements are sent to Sensor 2, as a *Bayesian network fragment*, as illustrated by Figure 5c.

In general, the *a posteriori* joint probability distribution will have more “dependency” or “correlation” among random variable or vector nodes. When the fragments are sent from Sensor

1 to Sensor 2, some of those correlation, represented by arcs, may be taken out as a part of approximation. We propose to use the *Bandwidth Agile Situation Dissemination (BASD)* technique, developed by IET and OSU, when this information is transferred between two sensor nodes. Detailed discussions of the application of BASD to the distributed data fusion will be described in Section 4.3, while the technical details of the BASD itself will be found in Appendix C. After the Bayesian network fragment illustrated by Figure 5c is transferred into Sensor 2, the fragment is used as the *a priori* state information by Sensor 2 and subsequently updated by the Sensor 2 measurements, as illustrated in Figure 5d.

3 DATA FUSION DISTRIBUTION LEVELS

This section defines and describes the three levels of distributed data fusion. The concept is based on the physical reality of the micro DADS and the sensor element capability.

3.1 Intra-Sensor Data Fusion

As mentioned in the previous section, each micro DADS sensor node, *i.e.*, a sensor element, assumed to have

- 1) an *acoustic sensor element* that measures the received sound pressure (the wide-band mode) and the frequency spectrum (the narrow band mode), and
- 2) a *magnetic sensor element* that measures a three-dimensional magnetic field.

The acoustic sensor element is capable of target detection that is binary information, positive or negative, about the range and the source level sound pressure implicated by the detection probability function, the received base frequency, and the additional acoustic spectrum structure that constitutes an acoustic signature useful for target classification. The time-series data fusion of the Doppler frequency may produce the CPA range and time, under a certain conditions.

A single measurement by the magnetic sensor element is capable of determining a three-dimensional manifold in the six-dimensional (or possibly lower-dimensional) target state consisting of the three-dimensional target position and the three-dimensional magnetic moment vector. Since the detection range by a magnetic sensor is generally very limited, the detection itself provides relatively good target localization. The time-series data fusion may determine at least higher dimensional sub-state estimate, although the observability dimension is not clear.

Finally it is possible to fuse information from the two sensor elements, acoustic and magnetic within a single sensor element. The most important data fusion is the detection fusion. In the current DADS implementation, an AND logic is used for target detection, and the simultaneous detection by an acoustic sensor and a magnetic sensor (both implemented by arrayed sensors) initiates a track.

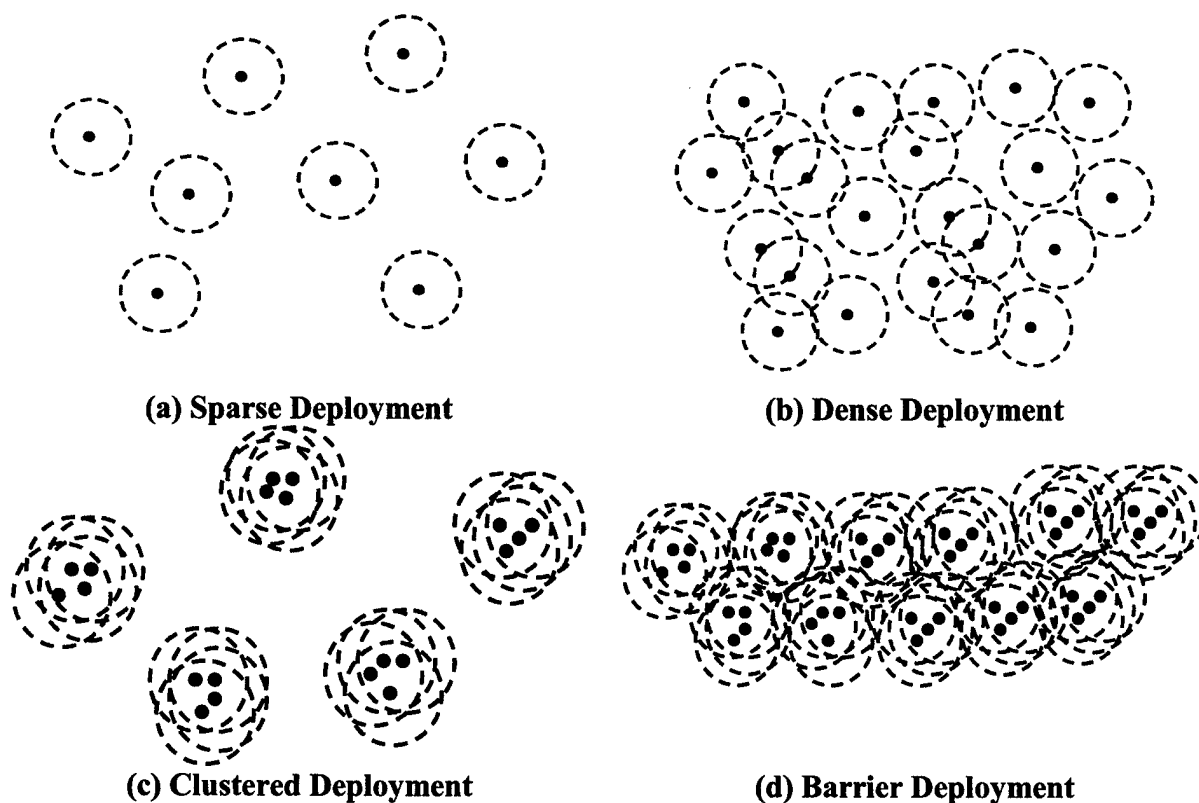


Figure 6: Micro DADS Deployment Patterns (Field Control)

Figure 6 shows some of possible micro DADS deployment patterns. In these figures, each filled circle represents a mini DADS sensor node and the dotted circle around it represents acoustic detection range. The sparse deployment pattern (Figure 6a) is similar to the one used in the current DADS implementation. If we use this pattern for the micro DADS, since each sensor node is just a single sensor element, its data fusion capability is very limited as described earlier in this section.

One of the novel concepts associated with the micro DADS implementation is formation of clusters as illustrated by Figure 6c. In this concept, several sensor nodes, *i.e.*, sensor elements, typically three or four, are placed in a cluster within a rather small proximity. As shown in Figure 6c, this placement does not increase detection range (coverage). However, as discussed below, formation of a cluster may generate an equivalent effect of forming a sensor array, just like the linear sensor array attached for each of the current DADS sensor nodes, provided effective coherent signal processing is possible among those sensor elements in a cluster.

Theoretically, if we deploy numerous mini DADS sensor elements in a very dense way, as illustrated by Figure 6b, we may create the same capability utilizing high-speed communication

between neighboring sensor nodes. Although this concept may constitute a unique system architecture, it is less effective than a more *organized* structure, such as the clustered organization illustrated by Figure 6c. Each sensor cluster then can be used as a current DADS sensor node and we can rather expand many of the currently used data fusion technologies ([1] – [7]) rather straightforwardly. Figure 6d shows an example in which a collection of sensor clusters is deployed to form a *barrier*.

3.2 Inter-Sensor, Intra Cluster Data Fusion

In this report, we assume that the mini DADS sensor nodes are deployed in a clustered pattern so that there is a clear sense of *groups* among sensor nodes. A key technology that may support this clustered sensor concept is coherent processing between neighboring sensor nodes through high-speed wireless communications. The distance between such nodes should be small enough to enable high-speed communication yet large enough to provide geometrical diversity within each sensor cluster.

Coherent signal processing of the acoustic signal received by two acoustic sensor elements may generate a TDOA measurement at two different locations from a target acoustic source, as a phase difference between two signals. A TDOA is translated into the difference between the two distances of the target from the two sensor locations, and hence, a TDOA measurement will define a rotated hyperbola with thickness defined by measurement errors. Theoretically three sensors may generate two-dimensional target localization by the intersection of two hyperbolae, and three-dimensional target localization may be possible using four acoustic sensor elements with three high-speed communication links enabling coherent processing. On the other hand, the time-series data fusion of two or three sensor elements may be enough to generate a complete target state estimate, *i.e.*, three-dimensional position and velocity.

Simultaneous observations of the magnetic fields at two different sensor locations may produce estimates of the three-dimensional target location and the three-dimensional target magnetic moment. This is a counterpart of the coherent processing of the acoustic signals. However, since we only need simultaneous observations and exchanges of those measurements between sensor nodes in order for the group of nodes to share this information, needs for high-speed communication is much less than that for the acoustic coherent processing. However, very small

detection range by magnetic sensor elements makes the use of such processing rather less significant, which makes the possibility of high-speed communication for acoustic coherent signal processing relatively crucial for the micro DADS concept.

On the other hand, fusion of target detection and classification information among clustered sensor elements can be performed with significantly less communications requirements. Using multiple sensor elements, performance of acoustic and magnetic target detection, as well as target classification, should be enhanced. As for target detection, for example, a desired detection probability can be obtained with lower false alarm rate, or a desired false alarm rate is obtained with higher detection probability. At this point, as far as effective target classification is concerned, benefit of data fusion is not very clear among clustered sensors, since the information pertaining target classification may be largely redundant among clustered sensor elements.

3.3 Inter-Cluster, System-Wise Data Fusion

Using the cluster concept, we may consider each sensor cluster as a full functional DADS node (an equivalent of a current DADS sensor node). In other words, when each sensor cluster is designed and placed in an effective way, it may generate decent two or three dimensional target localization in a rather short time, while retaining the same ability for target classification. However, target range in which sufficient target localization is possible may be a function of how well acoustic coherent processing can be performed through high-speed acoustic communication, as well as inter sensor geometry within a cluster.

Speed requirements for the communication between sensor clusters should be comparable to that of the current DADS implementation. Considering anticipated progress in the underwater acoustic communication technology (tele-sonar technology), any reasonable requirement will be satisfied for the future DADS concepts. With each sensor cluster organized to act like a single autonomous current DADS sensor node, the micro DADS, as a whole system, can perform many possible distributed data fusion functions coupled with many possible distributed resource management schemes, as documented in [1] – [7].

For example, a collection of clustered sensors can be placed to form a barrier to protect a port or a naval facility in a littoral area. Or it may be placed in a strategic location to perform “data fusion in transit,” by tracking a target from sensor cluster to sensor cluster, handing over the essential information, location, course, speed and target classification as the target transits through a field of sensor clusters as illustrated in Figure 7.

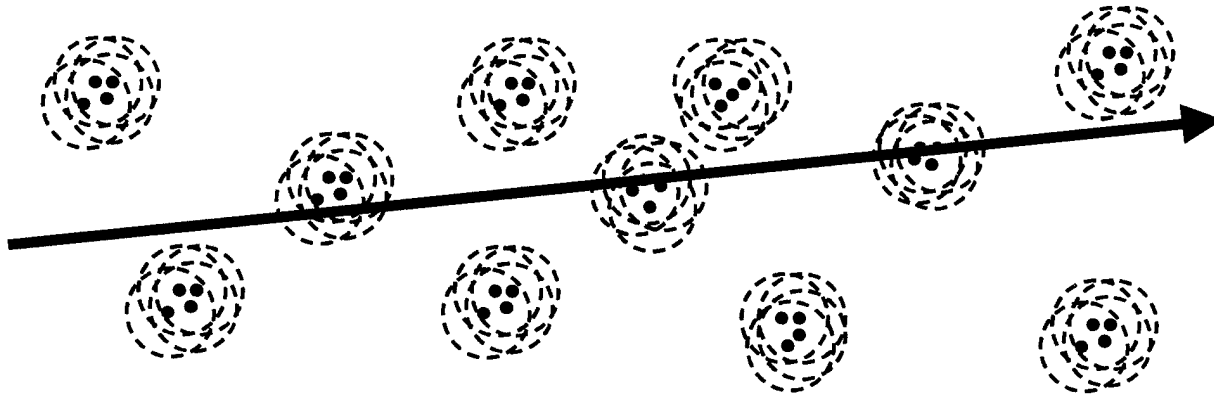


Figure 7: Data Fusion in Transit

In this figure, a transiting target is represented by a long arrow, indicating that three sensor clusters detect the target. A target track will be initiated when it hits a detection range of a sensor cluster, and subsequently, the target state estimates, as well as the target classification, will be handed over to the neighboring sensor clusters, depending on the target state prediction. Meanwhile, through inter-cluster communication, the target information is relayed to the surface communication gateway to connect the external tactical networks, through an RF connection. The sensor management, notably threshold control may be autonomously performed from cluster to cluster, or more centrally from a “master” node like the one in the current DADS concept.

4 DISTRIBUTED DATA FUSION SOLUTIONS

This section describes techniques that can be utilized to realize the distributed data fusion concept outlined in the previous section. We will use rather simplistic mathematical models to illustrate the multi-level data fusion technology as applicable to the future DADS concept.

4.1 Target Detection

We consider the *Neyman-Pearson* approach for the target detection ([32]). According to this approach, a binary hypothesis testing problem is considered: a hypothesis H_1 assuming that the received signal originates from a target, versus the null hypothesis H_0 assuming the received signal is only noise with no contribution from a target. The Neyman-Pearson criteria compare the likelihood ratio or the generalized likelihood ratio, with a given threshold to determine the detection, while the threshold is fixed according to the tolerable false alarm rate.

4.1.1 Single-Sensor Acoustic Detection

Using the model (1) in Section 2.3.1, and assuming the acoustic noise n_s is a zero mean Gaussian variable with a given standard deviation σ_s , the simple thresholding on the received sound pressure level y_s , we have

$$\begin{cases} P_D(r, s) = \text{Prob}\{y_s \geq \bar{y}_s | r, s, H_1\} = \text{erfc}\left(\frac{\bar{y}_s - (s/r^2)}{\sigma_s}\right) \\ P_{FA} = \text{Prob}\{y_s \geq \bar{y}_s | H_0\} = \text{erfc}(\bar{y}_s / \sigma_s) \end{cases} \quad (5)^2$$

where r is the target range, s is the source-level sound pressure, and \bar{y}_s is the threshold on the received sound pressure y_s . The decision rule is: conclude hypothesis H_1 assuming a target if $y_s \geq \bar{y}_s$, otherwise conclude the null hypothesis H_0 .

² By erf and erfc , we mean the error function and the complementary error function defined as $\text{erf}(x) = \int_{-\infty}^x e^{-\xi^2/2} d\xi / \sqrt{2\pi}$ and $\text{erfc}(x) = 1 - \text{erf}(x)$.

Figure 8 shows the detection probability as a function of the target range r , normalized by the expected source-level sound pressure s , the false alarm probability, and the noise level σ_s . By defining the detection as a conditional probability conditioned by the target's state, it is possible to update any target state distribution by the fact that

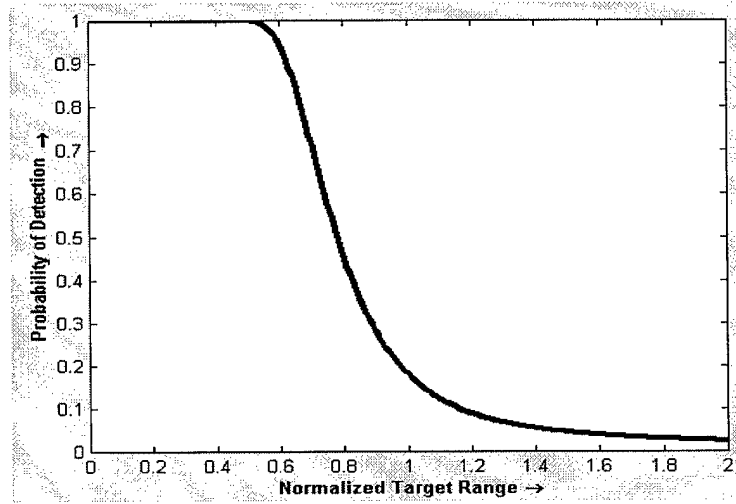


Figure 8: Acoustic Detection Probability

it is detected (*positive information*), as well as the fact that it is *not* detected (*negative information*). For example, the *a priori* target state distribution with the density function \bar{p} can be updated by the positive or negative information as

$$\left\{ \begin{array}{l} \hat{p}(x) = \frac{P_D(x)\bar{p}(x)}{\int P_D(\xi)\bar{p}(\xi)d\xi} = \frac{P_D(r,s)\bar{p}(x)}{\int P_D(r,s)\bar{p}(\xi)d\xi} \quad \text{or} \\ \hat{p}(x) = \frac{(1-P_D(x))\bar{p}(x)}{\int (1-P_D(\xi))\bar{p}(\xi)d\xi} = \frac{(1-P_D(r,s))\bar{p}(x)}{\int (1-P_D(r,s))\bar{p}(\xi)d\xi} \end{array} \right. \quad (6)$$

The pair (r, s) of the target range and the source-level sound pressure in (6) is a function of the state vector x or ξ , as $r(x)$ and $s(x)$, or $r(\xi)$ and $s(\xi)$.

4.1.2 Multiple-Sensor Acoustic Detection

With a rather large number of “small” detection-only acoustic elements, as in the densely distributed micro DADS concept (as opposed to the cluster concept), or the “sono-buoy-field” concept, the simultaneous detection and non-detection, or in other words, positive and negative information, may be combined to produce useful information. The quality of information gained in this way may depend on the detection probability characteristics and the placement of the acoustic elements. The updating by such simultaneous detection and non-detection can be expressed as a product of detection probabilities $P_D(s, r_i)$ or $(1 - P_D(s, r_i))$, where s is the target's source-level sound pressure, and r_i is the target range from the i -th sensor.

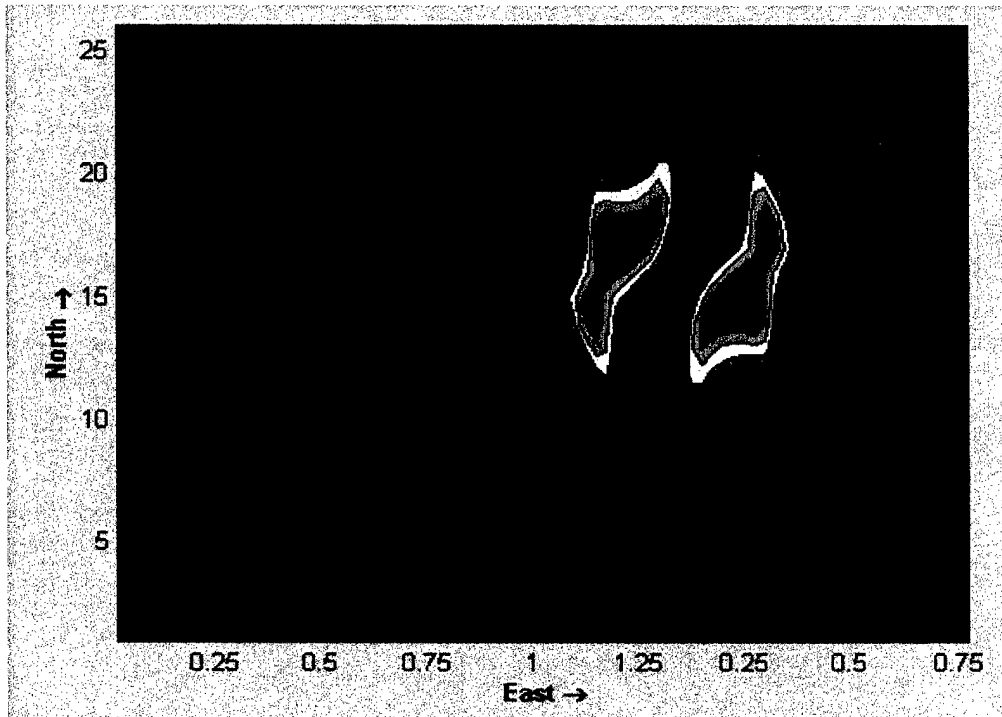


Figure 9: Multiple-Sensor Acoustic Detection

Figure 9 illustrates the possibility of target localization through acoustic detection information. In Figure 9, nine sensors are placed in a staggered array, either by green circles or red squares, then each sensor detects or does not detect a target. The target location is depicted by the purple circle. The north-east distance is normalized by the detection range and the expected source-level target sound level.

As illustrated in Figure 9, target localization by this positive negative information combination is rather limited. Apparent non-Gaussian, potentially multi-modal probability distributions may not be represented or used in any useful way. It is for this reason, coherent acoustic processing that may provide much better information (that supports the sensor cluster concept) may be essential for successful implementation of the micro DADS concept.

4.1.3 Single-Sensor Magnetic Detection

As shown in Section 2.2.2, a magnetic sensor element can measure the three-dimensional magnetic vector field as a function of the three-dimensional target magnetic moment vector and the three-dimensional vector from the sensor location to the target location, contaminated by the background and other noises. Appendix B shows how the target detection problem is formulated

as a hypothesis testing problem on the chi-square statistics $\chi^2 = y_m^T R_m^{-1} y_m$ of the three-dimensional magnetic field measurement y_m , with the 3×3 covariance matrix R_m of the noise. The noise can be considered as a zero-mean Gaussian random vector after removing any significant locally persistent background magnetic field.

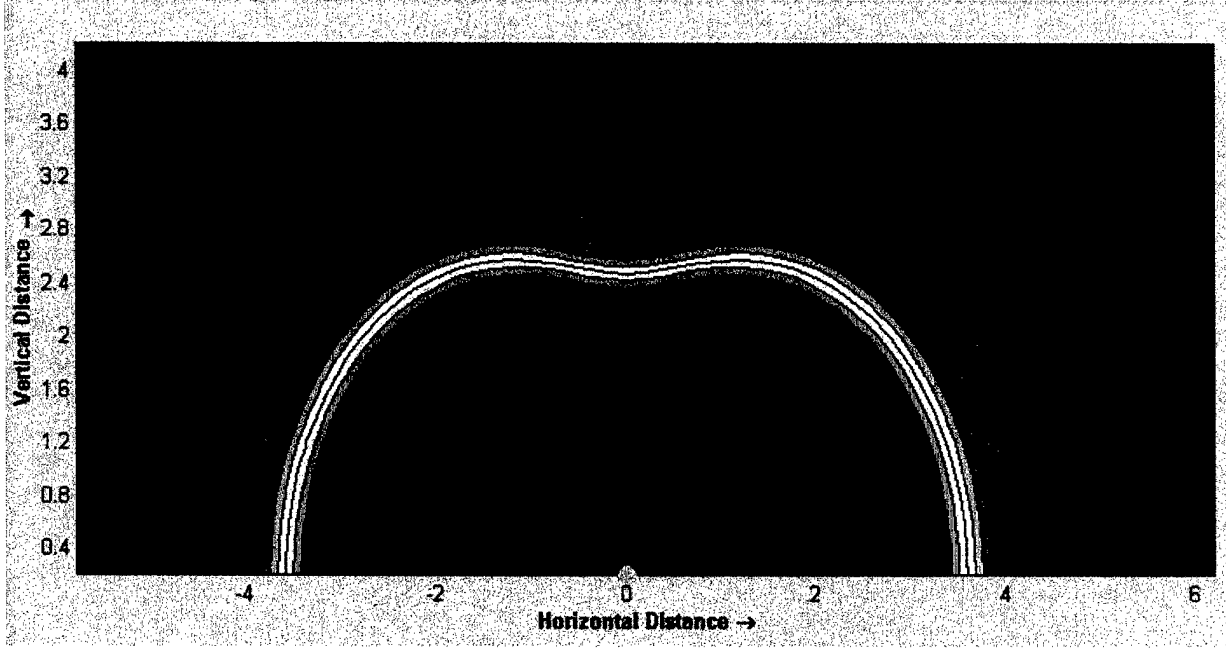


Figure 10: Target Localization by Single Magnetic Detection

As shown in Appendix B, the Neyman-Pearson approach can be applied to define a single-sensor detection process. Figure 10 shows the target location estimated from the condition that the target is detected as $\chi^2 = y_m^T R_m^{-1} y_m \geq \bar{\chi}^2$, assuming that the target's magnetic vector moment is aligned to the horizontal direction of the figure. The distance in the figure is normalized by the source magnetic moment strength. The magnetic sensor element location is shown by the green circle.

4.1.4 Multiple-Sensor Magnetic Detection

As shown in Section 4.1.2, negative and positive information can both be used for target localization using the probability detection function, $P_D(\Delta u, m) = \text{Prob}\{\chi^2 \geq \bar{\chi}^2 \mid \Delta u, m, H_1\}$, where $\Delta u = u_T - u_S$ is the three-dimensional vector from the sensor location u_S to the target location u_T , and m is the target magnetic moment vector, using the model described in Section 2.2.2.

As in the case of acoustic sensors described in Section 4.1.2., the simultaneous detection by multiple magnetic sensors may provide accurate information about the target location. For example Figure 11 shows the probability distribution of the target location conditioned by the fact that it is simultaneously detected by two magnetic sensors represented by the green circles.

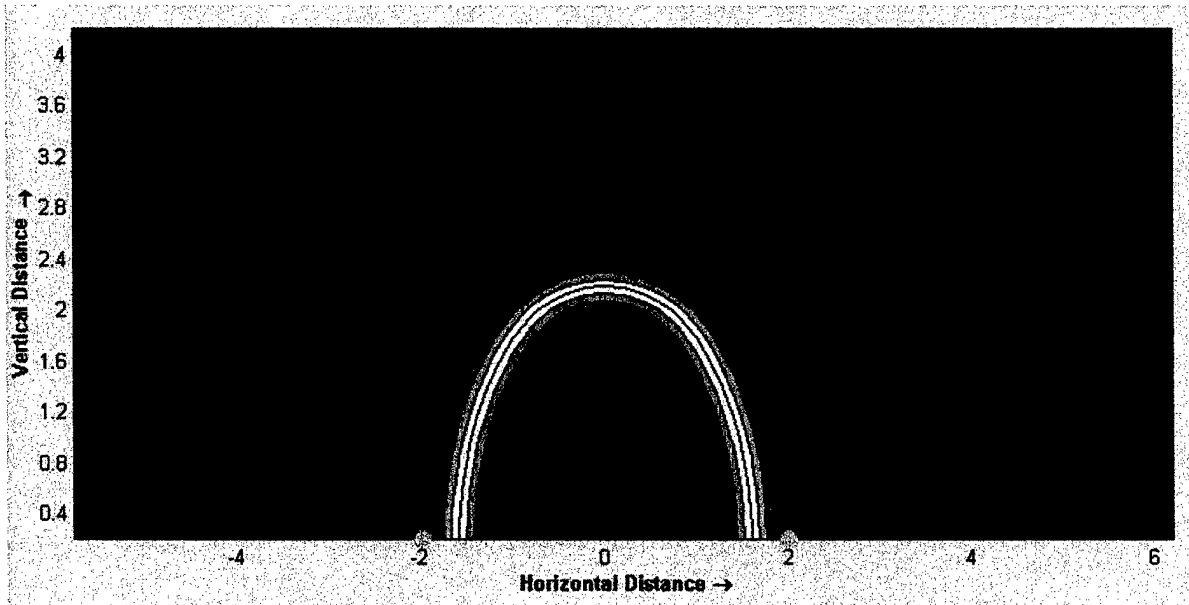


Figure 11: Target Localization by Two Simultaneous Magnetic Detections

The negative information is represented by $(1 - P_D(\Delta u, m))$. Figure 12 illustrates the target localization by the negative information.

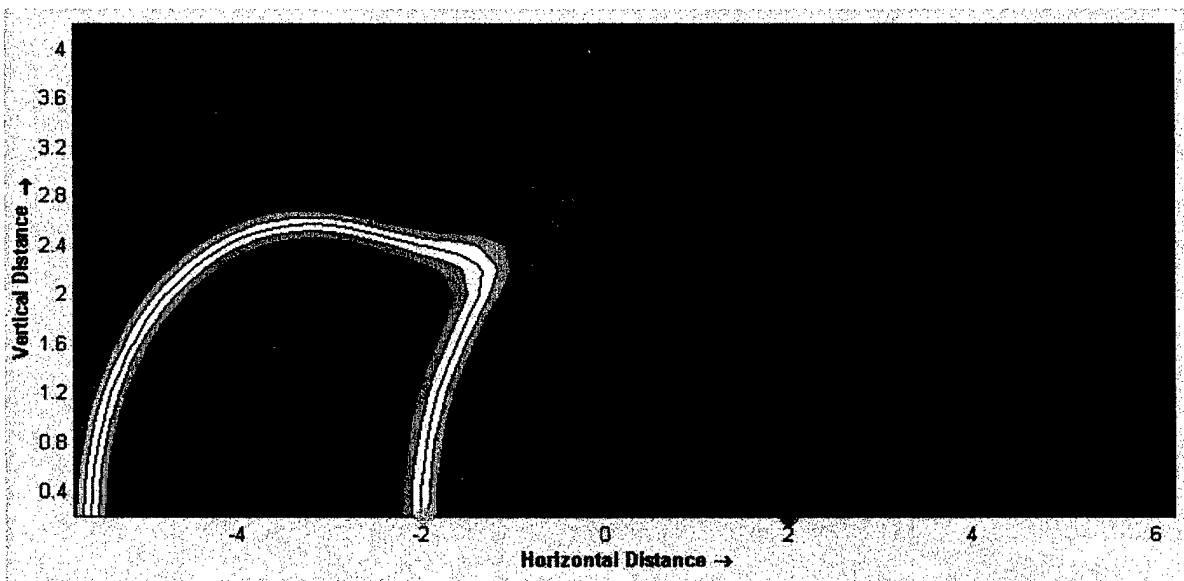


Figure 12: Target Localization by Positive and Negative Information from Two Magnetic Sensor Elements

In Figure 12, we assume the left sensor element (the green circle) detects a target while the right sensor element (the gray circle) does not detect it.

4.1.5 Multiple-Sensor Distributed Detection

In the previous sections, it has been discussed how to combine information gained only by the fact that a certain sensor detects a target using its own local rule, notably using the Neyman-Pearson method, or by the fact that a certain sensor looks for a target but does not detect one. In the case of the latter, *i.e.*, the positive information, a magnetic sensor will provide numerical information that is much richer than that provided by a single acoustic element, as discussed later in Section 4.2. As usual, any negative information only gives limited information. In any case, the issue of usefulness of the information aside, the binary information, *i.e.*, detection or no-detection, can be used for target localization, using the detection probability function.

On the other hand, as shown in Appendix B, target detection can be performed collectively by a set of similar or dissimilar sensors. In order for that to be effective, however, it is necessary for each sensor to exchange the local likelihood ratio as well as some sufficient statistics of the target state (or the location) under the hypothesis that each signal contains the signal component from a target. The communication load necessary for this can probably be considered as the minimum. Particularly, it would be nothing compared with the communication load required by coherent acoustic processing. Since, as information about the same target, information gained from the remote location is probably useless. When put together, distributed detection is probably only meaningful among sensor elements within a sensor cluster, *i.e.*, for inter-sensor, intra-cluster data fusion.

Distributed detection can be viewed as a *pre-detection data fusion* method. An alternative is to fuse the detection decisions that are made locally, independently, according to the “local” rules. We should point out that, as an alternative to distributed detection and data fusion of binary data (detection or non-detection), there is a body of theory that views the joint detection process as a *team problem*, where the communication among the members (sensor elements) is limited to be binary, known as *decision fusion* ([33]). While these issues should be studied more for their potential contribution to the micro DADS concept, because of its complexity, utility of pre-detection fusion and decision fusion does not appear promising.

4.2 Target Localization and Tracking

One of the significant challenges posed by the micro DADS concept is that, because each DADS sensor node is essentially a single sensor element (although, as we assume, it may have an acoustic as well as a magnetic sensor element), unlike a current DADS sensor node, it cannot produce any complete target localization by itself. Therefore, it is necessary for each node to fuse data either temporally (time-series integration) or spatially (distributed data fusion). This section describes several possibilities of such data fusion at different levels of data fusion distributed processing.

4.2.1 Acoustic TDOA Localization and Tracking

It is well known ([9] – [11]) that though coherent signal processing, it is possible to estimate the difference of the time when a certain signal from a target arrives at a sensor from the time when the same signal arrives at another sensor located nearby but with certain spacing [9]. This TDOA, Δt , is directly translated into the difference $\Delta r = r_1 - r_2$ between the range r_1 of the target from one acoustic sensor and the range r_2 of the target from another sensor. Given three-dimensional sensor locations of the two sensors and this difference $\Delta r = r_1 - r_2$ of the ranges, it is well known that a surface on which the target is located can be determined.

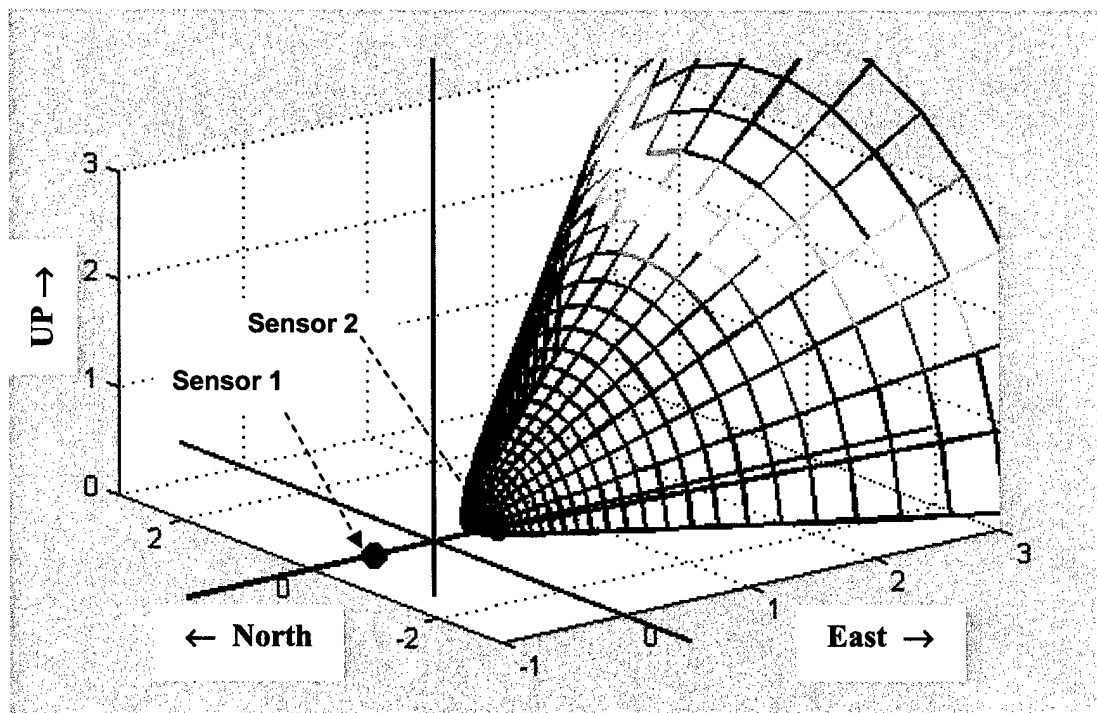
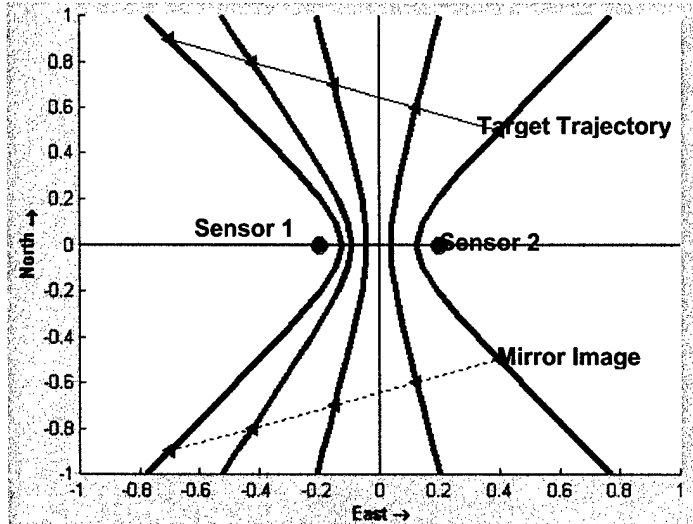


Figure 13: Hyperbolic Surface Determined by TDOA

As illustrated in Figure 13, a noiseless TDOA measurement corresponds to a hyperbola defined by the distance-difference from the two sensors (as its two foci) on a horizontal plane containing the two sensors, and to the two-dimensional surface in the three-dimensional space obtained by rotating the hyperbola along the line defined by the two sensor locations. Measurement and processing errors will provide some “skin depth” to this surface. An error in the TDOA estimate translates to the errors that increase as the distance from the sensors increases.

**Figure 14: TDOA Tracking**

The temporal integration, or data fusion, of TDOA measurements, coupled with a target dynamic model, such as the Constant Course and Speed model, may provide us with the four dimensional (2-dimensional position plus 2-dimensional velocity) target state estimate, as illustrated in Figure 14.

This batch-processing algorithm may generate good four-dimensional target state estimates when the target depth is known, *e.g.*, through magnetic sensor element measurements. However, as shown in Figure 13, the state estimation in this method clearly generate a “mirror” image, just as a mirror bearing from an acoustic line array. This ambiguity must be resolved using data fusion with other sensor nodes.

On the other hand, if we use a cluster of three acoustic sensor elements, we may have two independent TDOA measurements, from which we can localize a target position within a plane. As shown in [9], there is an analytic method for exactly determining the intersection of two hyperbolae. This means that, using three acoustic sensor elements (nodes) in a cluster in such a way that coherent signal processing produces two simultaneous TDOA's, it is possible to

produce a two-dimensional target position estimate instantaneously. This hyperbola intersection algorithm is illustrated by Figure 15.

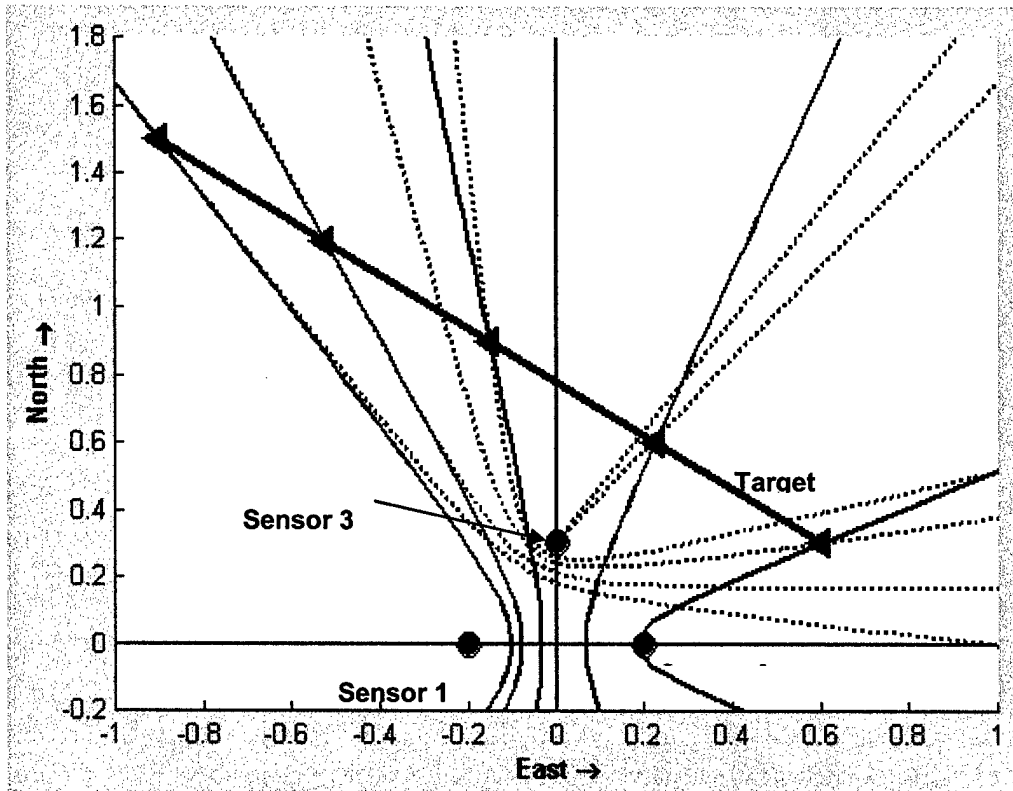


Figure 15: Hyperbola Intersection Method

In Figure 15, the TDOAs between Sensors 1 and 2 are depicted by the green hyperbola, and those from Sensors 2 and 3 by the dotted blue hyperbola. In Figures 13 to 15, the location scale is normalized by the sensor location. In general, from the viewpoint of the localization accuracy, it is desired to have large distances between sensor elements within detection range. However, as a micro DADS concept, the more distance between two sensor elements, the more expensive the high-speed wireless communication to enable the coherent processing to produce good TDOA estimates. This tradeoff between estimation accuracy and communication requirements is one of the studies that may be crucial for the micro DADS concept.

Figure 16 illustrates the accuracy of the hyperbola intersection target localization algorithms by showing the measurement likelihood function obtained by the two TDOA measurements. For this figure, the scale is normalized by the TDOA measurement accuracy (translated to the distance difference).

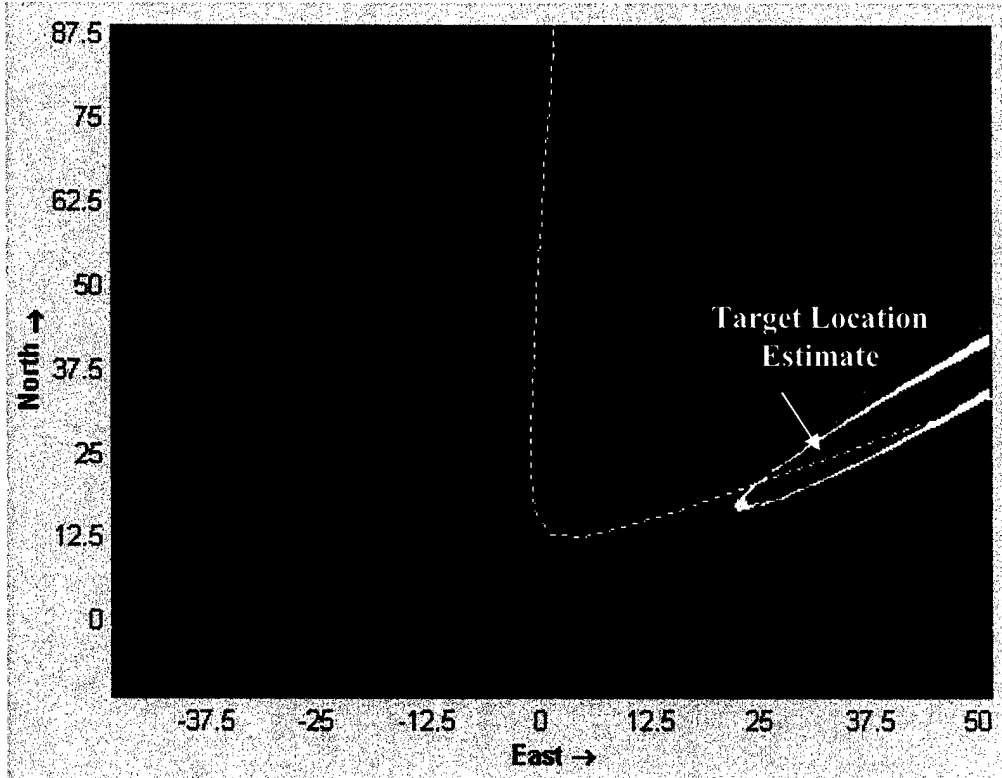


Figure 16: Likelihood Function of Hyperbola Intersection Localization Algorithm

In Figure 16, two hyperbolae produced by the sensor pairs, 1 and 2, and 2 and 3, are shown and the target location estimate is obtained as the intersection. The shape of the likelihood function shown in Figure 16 shows that the estimation error increases as a function of range. Indeed, estimates obtained this way may resemble bearing measurements more than a Gaussian target localization estimate. This indicates the need either for more acoustic elements within a sensor cluster or for inter-cluster data fusion (which may be equivalent of bearing intersection target localization).

4.2.2 Acoustic Doppler Tracking

As described in Section 2.2.1, each single acoustic sensor element, *i.e.*, each single micro-DADS node, may measure Doppler frequency, provided a narrow-band acoustic processing with enough accuracy and enough signal-to-noise ratio in the environment ([14]). Figure 17 illustrates a possible frequency observation around the CPA event. The scales are normalized by the CPA time t_{CPA} , the CPA range r_{CPA} , the target speed V , the source frequency f_s , and the speed of the

sound in the medium (the water) c , assuming the Constant Course and Speed (CCS) behavior of the target around the CPA event.

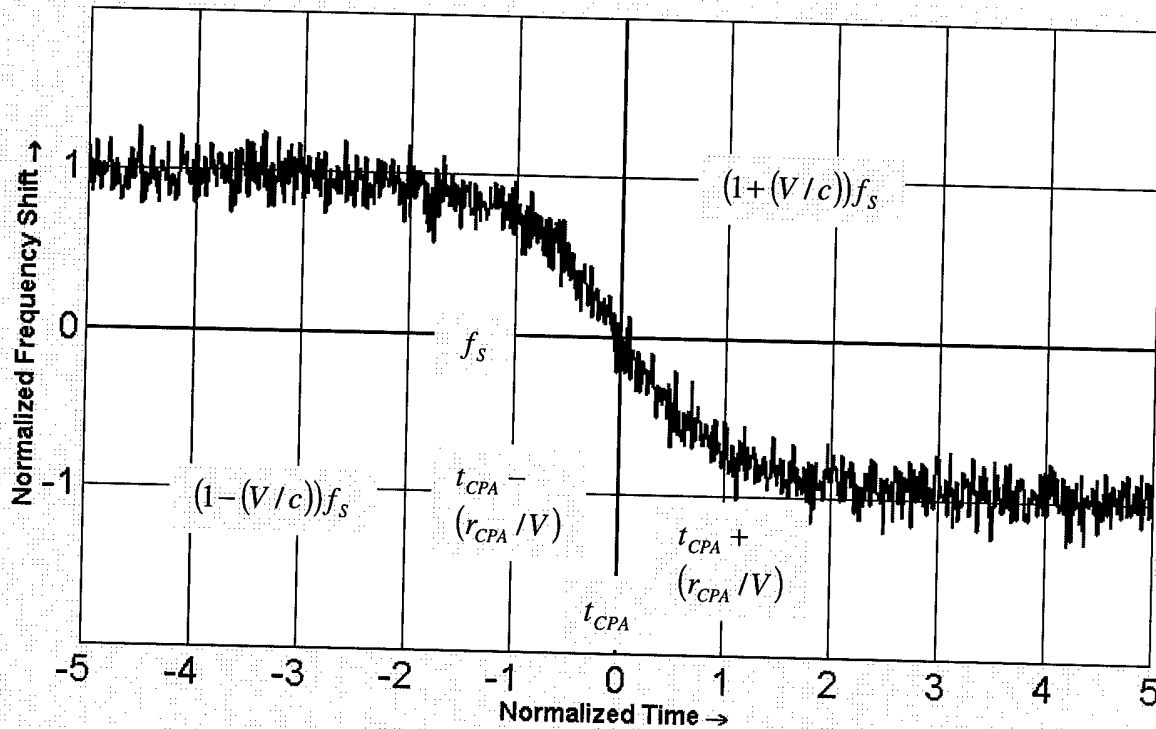


Figure 17: Doppler Frequency Measurements

Potentially, by observing this CPA event by a single acoustic element, we can estimate the CPA time t_{CPA} , the CPA range r_{CPA} , the target speed V , and the target source base frequency f_s . The received sound pressure measurement is generally proportional to the source-level sound pressure s and the square of the inverse of the target range, $1/r^2$, and may provide additional information to estimate the CPA time, as illustrated in Figure 18.

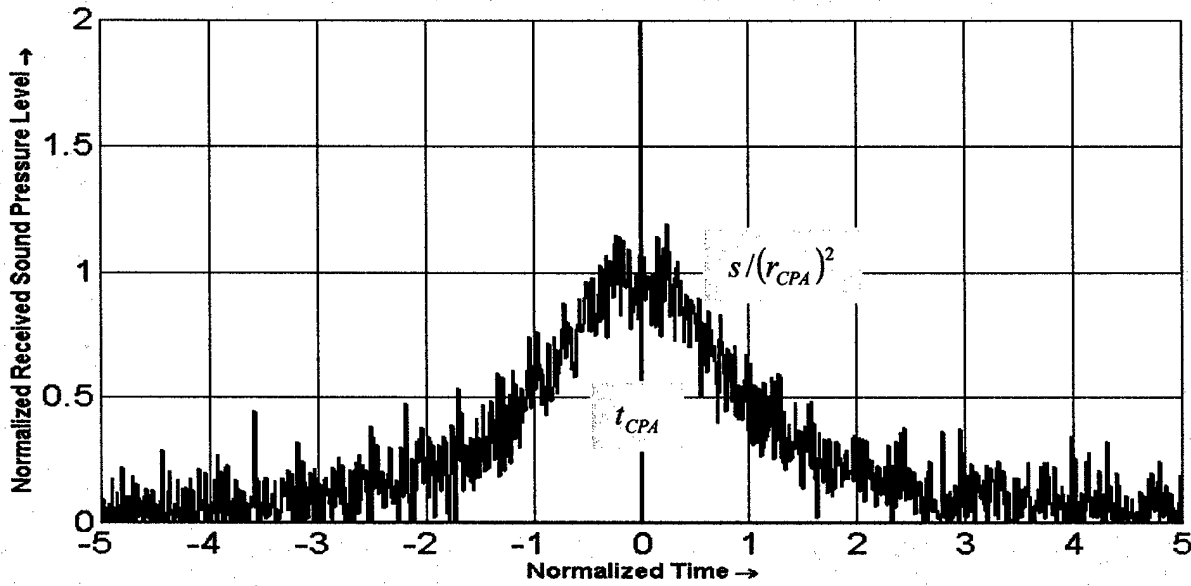


Figure 18: Received Sound Pressure Measurements

For each CPA event observed by a single acoustic element, there is one unobservable component of the target state, which is the target course (assuming a CCS target). The course may be determined by the two CPA events observed by two acoustic sensor elements, as illustrated in Figure 19.

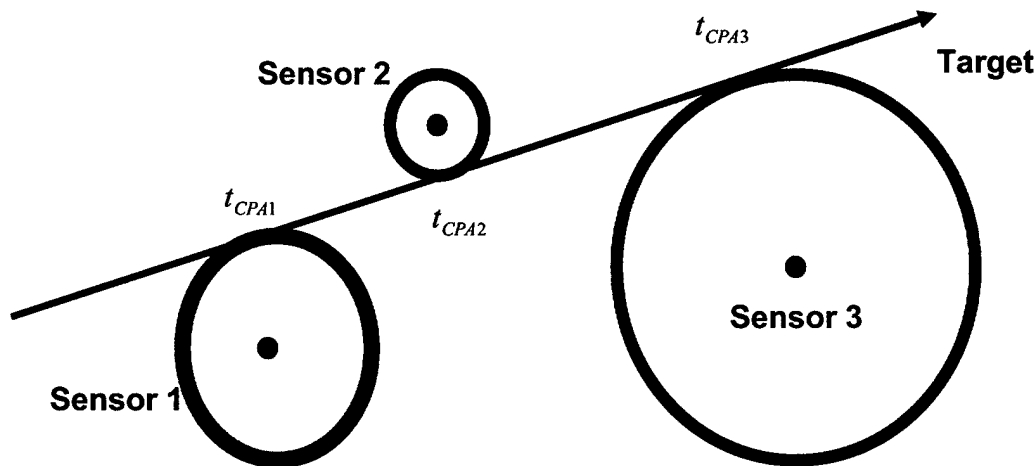


Figure 19: Multiple-Sensor Doppler Tracking Data Fusion

In this figure, three CPA events at t_{CPA1} , t_{CPA2} , and t_{CPA3} , by three acoustic elements, Sensors 1, 2 and 3, are illustrated. The blue ring associated with each sensor represents the CPA range

estimates. Together with other estimates on the CPA time t_{CPAi} , the target source frequency f_s , the target velocity V , and the source level sound pressure s , it is possible to fuse estimates from multiple sensors to generate a full state distribution.

This Doppler tracking provides us with a flexible data fusion method. Since it does not require any coherent processing, required communication among sensor nodes is minimal. Hence this method for distributed data fusion can be applied to both the inter-sensor, inter-cluster (Level 2) and the inter-cluster (Level 3) distributed data fusion. Possible drawbacks include vulnerability to the target maneuvering, which prevents the data fusion illustrated in Figure 19 in a relatively short time period.

4.2.3 Magnetic Localization and Tracking

As shown in Section 2.2.2, a single magnetic sensor element measures a three-dimensional magnetic field. As shown in Section 4.1.3, the magnetic field loses its strength in proportion to the cubic power of the range, and because of the usual background magnetic field, the detection range is very small ([1],[12],[13]). However, within the detection range, the magnetic measurement may provide accurate target information. As shown by eqns. (3) and (4) in Section 2.2.2, a single observation is a three-dimensional magnetic field measurement, while the target state component involved in eqn. (3) is the three-dimensional target location plus the three-dimensional target magnetic vector moment, *i.e.*, a six-dimensional sub-state vector. Therefore, in order to determine the state with enough accuracy, we need to fuse data either temporally or spatially. Using the micro DADS sensor cluster concept, however, it is relatively easy to obtain decent estimates of the target three-dimensional location and the three-dimensional target location with simultaneous measurements at a single time instance.

Simultaneous magnetic field measurements can be expressed as

$$y_m = \begin{bmatrix} y_{m1} \\ y_{m2} \end{bmatrix} = \begin{bmatrix} G(\Delta u_1) \\ G(\Delta u_2) \end{bmatrix} m + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} \quad (7)$$

where y_{mi} is the three-dimensional magnetic field observation by Sensor i , Δu_i is the three-dimensional vector from the location of Sensor i to the target, m is the three-dimensional target magnetic vector moment, n_i is the background magnetic noise, and $G(\cdot)$ is the matrix-valued

function defined by eqn. (4) in Section 2.2.2. Hence, it may be possible to obtain joint estimates of the target location and the target magnetic moment m from the six dimensional measurement y_m . Figure 20 illustrates such data fusion.

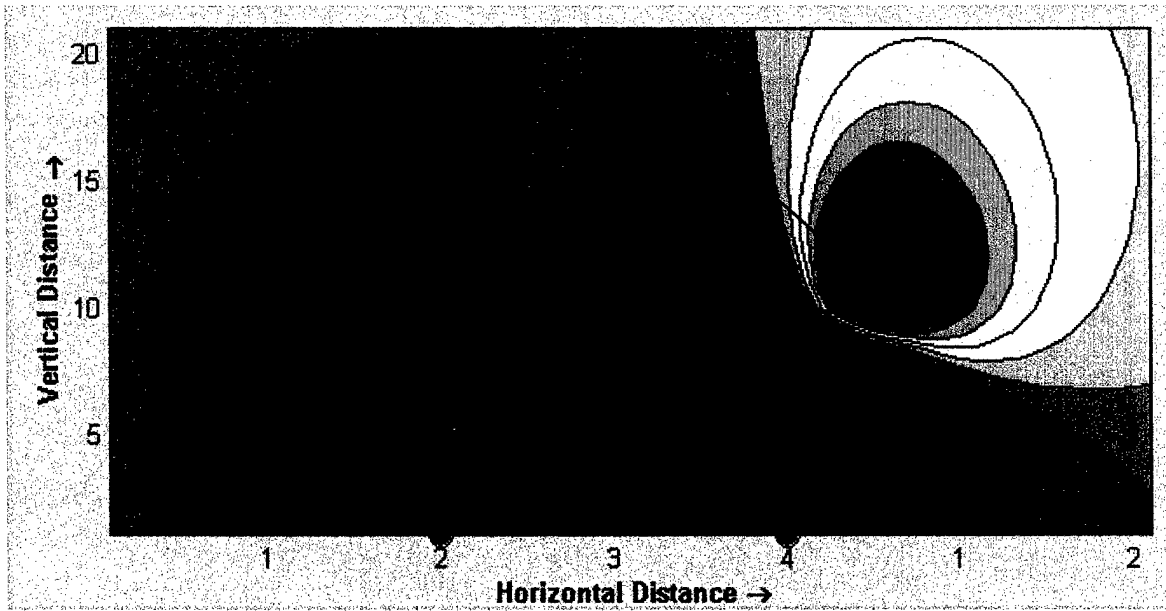


Figure 20: Two-Sensor Magnetic Localization (Vertical)

Figure 20 shows an example of the likelihood function within a vertical plane containing two sensor locations (on the bottom of the ocean). The scales are normalized by the sensor spacing, and the noise level is assumed to be 0.1 of the magnetic source signal level (SNR = 6dB). Figure 21 shows the same likelihood function in a horizontal plane which contains the target. The sensors are not in the plane shown by Figure 21 (They are on the bottom).

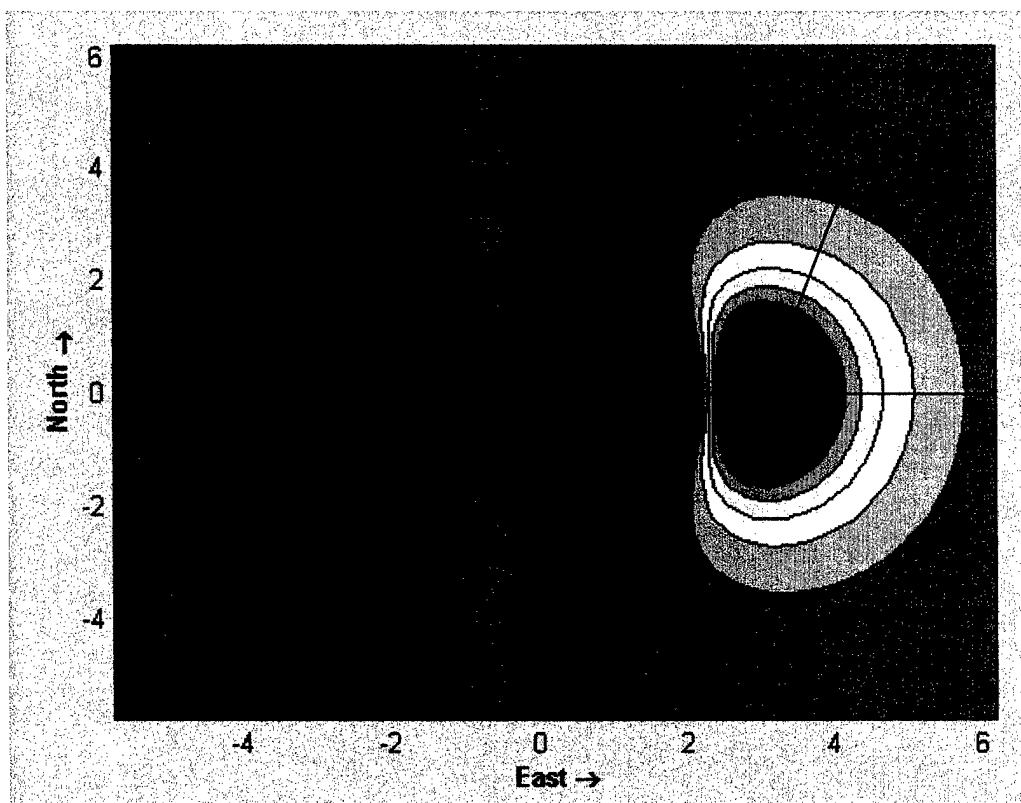


Figure 21: Two-Sensor Magnetic Localization (Horizontal)

Like Figure 20, the scales of Figure 21 are normalized by the sensor spacing. It is very interesting to see the non-Gaussian *a posteriori* distribution resulting from the magnetic field observation. The rather complex contours shown in Figures 20 and 21 indicate that the nonlinear problem to obtain a “good” estimate for the three-dimensional target location may not be easily solved. Depending on situations, a very powerful nonlinear, non-convex optimization technique may be required.

This difficulty may be overcome by fusing data that are gathered temporally (time-series) or spatially (more sensor elements). A significant fact is, however, this multiple-sensor-element magnetic localization method does not require any coherent processing. However rather very small detection range makes only the intra-cluster data fusion possible. Since the three-dimensional localization is possible, the information is very useful for the inter-cluster, system-wise distributed data fusion.

4.2.4 Inter-Cluster Distributed Tracking

In the earlier sections, it was shown that it is possible to configure a micro DADS sensor elements in such a way that they collectively form a set of current DADS sensor nodes with acoustic and magnetic arrays. As mentioned before, in order for this to be possible, we need wireless high-speed acoustic (or any other possible means) that enables coherent acoustic processing among adjacent sensor elements. In other words, it is possible to realize the current DADS sensor node functionality by a cluster of micro DADS sensor nodes, each of which has only a single acoustic sensor element and a single magnetic sensor element.

In other words, a micro DADS concept may make a sensor cluster that may function as a single sensor node of the current DADS concept. If this concept is realizable, all the data fusion concepts and the system control concepts that have been developed up until now ([1] – [7]) can be realized by the micro DADS concept. Assuming the coherent acoustic signal processing is possible through high-speed communication, it may be possible to obtain two or three dimensional target localization at a single time, although from time to time, target localization may resemble bearing measurements rather than positional information. If this is accomplished, many diverse techniques for tracking and data fusion ([15] – [28]) can be applied.

In particular, we should point out that a general solution to the multiple target tracking problem [30] and its extension to general distributed tracking problems [29] can be used to maintain optimality or near-optimality in distributed data fusion.

In distributed tracking or data fusion, the following decisions must be made by each distributed sensor node:

- (1) *Information Dissemination:* Depending on information accumulated by each sensor node (cluster), when, to whom, and what information should be transmitted. The transmission frequency may be an important decision depending on the communication and processing capacity of adjacent sensor nodes. In addition, decision should be made, depending estimates of a target, as to which node should be communicated (*e.g.*, the direction that the target moves).

(2) *Distributed Data Fusion*: In general, information sent from another sensor node (cluster) is aggregated information rather than raw data (measurements). Hence each node must be able to fuse local data and those foreign data to come up with local global information. Each node also must be able to remove informational redundancy if two sensor nodes repeatedly communicate to each other. If each node also communicates with some basic hypotheses, *e.g.*, detection hypothesis, association hypothesis, *etc.*, it is also necessary to maintain consistency by remembering the past informational states.

(3) *Situation Assessment*: Since overall communication speed within the overall system may be limited because of use of acoustic communication, some urgent information must be propagated to external nodes through an RF gate with much higher priority than other usual communication. This means each node (cluster) should be able to form a situation assessment by itself, to disseminate such information to the external node as soon as possible.

Distributed data fusion is also driven by target classification that can be performed, in large part, autonomously within a single (micro DADS) sensor node. That aspect will be discussed in the next section.

4.3 Dynamic, Distributed, Hierarchical Classification Using Bayesian Networks

4.3.1 Problem Statement

Formally, classification seems a simple problem. The task is to compute a probability over a set of possible target types, T , given a set of observable data, e . Given $P(e|T)$, $P(T)$, and a specific e^* that is observed, one can simply compute:

$$P(T|e^*) = P(e^*)^{-1} P(e^*|T) P(T) \quad (8)$$

More generally, if e can be broken into subsets as (e_1, \dots, e_n) that are independent given T , we have

$$P(T|e^*) = P(e^*)^{-1} P(e_1^*|T) P(e_2^*|T) \dots P(e_n^*|T) P(T) \quad (9)$$

While the above may be useful for dealing with multiple sensors, though, it does not solve several emerging problems:

1. Computational complexity of the above is linear in the number of possible target types.
This makes dealing with large numbers of types difficult.
2. Sensor evidence is rarely truly independent given target state.
3. The simple type model above is inadequate for dealing with increasingly richly structured information about a type (e.g., modal information).
4. The simple model above is inadequate for representing type-based temporal correlations (e.g. type-based behaviors)
5. The above does not provide for distributed cooperative classification unless information from multiple platforms can be modeled as independent given the type.

In this report we present an alternative formulation of classification as a *dynamic, distributed, structured* process.

- **Dynamic** – we will propose dynamically managing the number of types under consideration.
- **Distributed** – we will propose methods for multiple sensors to cooperate in classifying a platform in the presence of communication bandwidth limitations.
- **Structured** – we will propose a structured, hierarchical representation to manage the increasingly complex information about platform types.

Our proposal will be based on the use of Bayesian networks as a fundamental representational and communication element. A Bayesian network is a structured representation of probability information, especially useful for compactly representing information about large, highly structured domains. Use of Bayesian networks enables compact representation and efficient application of classification knowledge.

We begin with a discussion of single-site classification in section 4.3.2. In section 4.3.3 we explore issues in distributing the classification process.

4.3.2 Local Classification

4.3.2.1 Structuring Classification Knowledge for Representation and Application

There is a wide variety of knowledge potentially applicable to the classification task. This knowledge is relevant through its direct or indirect effect on observables. We have identified four varieties of knowledge about platforms:

- Knowledge of static physical characteristics
- Knowledge of dynamics
- Knowledge of operating modes (discrete parameters that, typically, induce correlations among physical characteristics)
- Knowledge of behaviors (typical or important sequences of operating modes)

Varying subsets of this knowledge may be relevant to differing classification tasks. We would like to represent this knowledge in a way that permits flexible application of only those parts needed.

In addition to structure in the classification knowledge for a single platform, there is structure in our knowledge about the relationship among platforms. “Family tree” or hierarchical structures are useful both for representing similarities among platforms, and also for economy of reasoning – they can be used to rule in or out entire sub-families of platforms by considering only a single hypothesis.

For the above reasons we propose a structured, hierarchical representation of classification knowledge. An example of such a representation is shown in Figure 22 below.

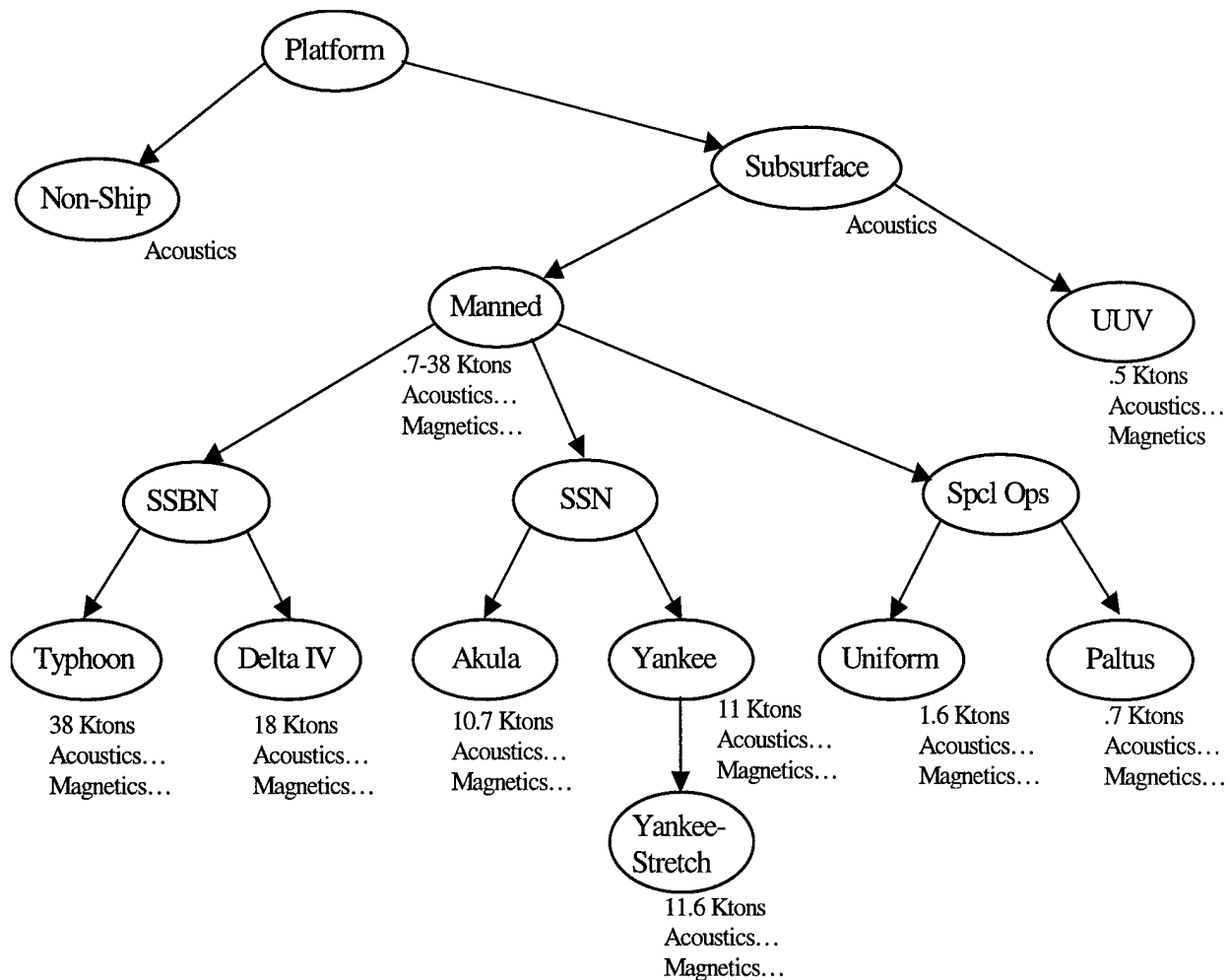


Figure 22: Hierarchical Representation Of Classification Knowledge

The above representation defines $P(e|T)$ for each of the leaf types in the hierarchy through the standard Object-Oriented programming language semantics of inheritance and overriding. Further details are beyond the scope of this document, see [34], [35] for more details.

The left side of Figure 23 below shows how a typical platform model might look at the graphical level (the middle of the diagram is a simplified acoustic conduction path model, including very simple models of acoustic and magnetic sensors. As can be seen from this model, in this discussion we are focusing on modeling for classification, so traditional state vector elements such as position, needed for tracking, are not included. We propose accounting for correlations among sensors through a model based approach that directly represents the dependence of sensor information on underlying shared platform parameters.

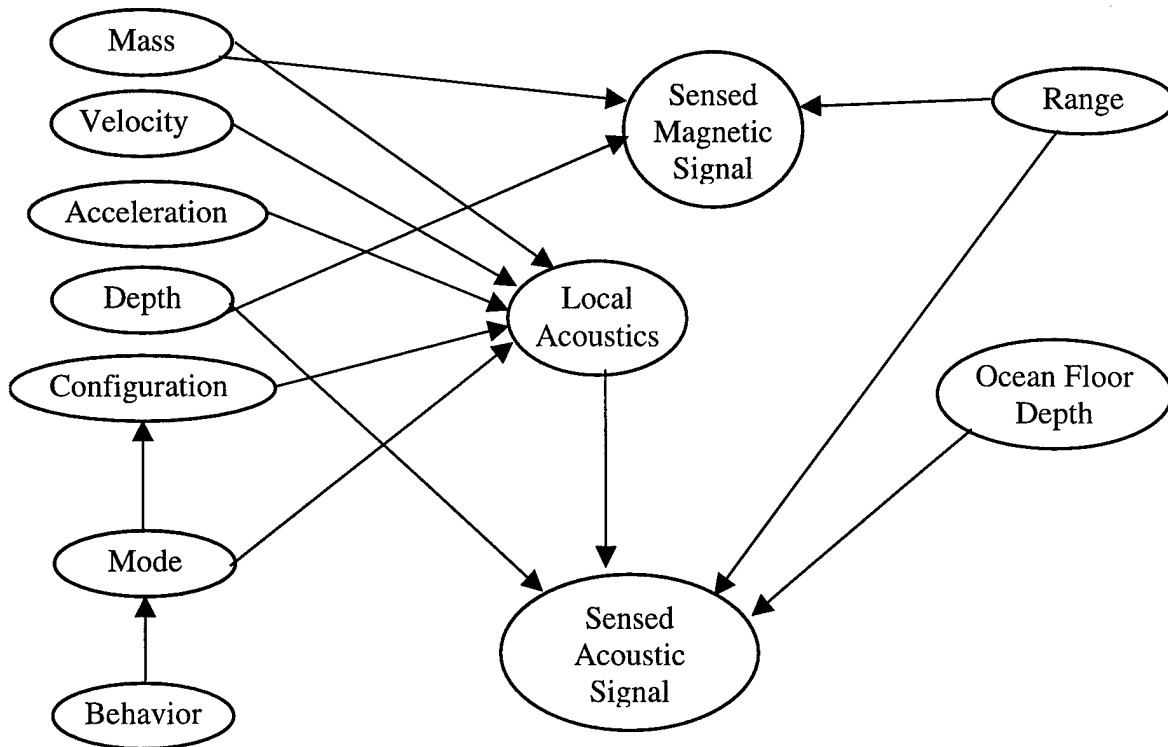


Figure 23: Simplified Typical Platform Model

The overall complete classification model, combining the above two, then, is as follows at the graphical level (again, this is simplified, and does not account for platform existence uncertainty. See [34], [35] for more details):

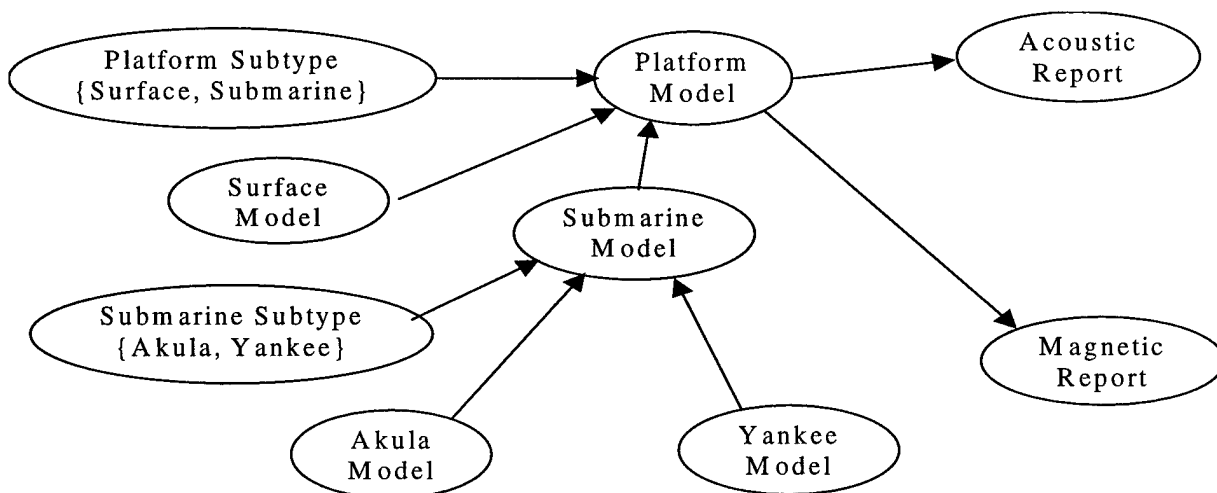


Figure 24: Sample Classification Tree for an Unknown Platform

In this model platform classification information is found in the "... subtype" nodes. Once evidence is asserted on the magnetic report and acoustic report nodes (actually multi-

dimensional), posterior probabilities of the *Platform Subtype* and *Submarine Subtype* nodes can be computed using standard Bayesian network inference algorithms.

4.3.2.2 Dynamic Classification

The above is a useful convention for organizing, representing, and computing with classification knowledge. However, we believe it necessary to go further and dynamically manage the classification space in much the same way that trackers dynamically manage tracks and data associations. At any point in time current posterior about a target to be classified can be represented using only a subset of the classification hierarchy. An example is shown in Figure 24 above. In that example, the platform in question is thought to be either a Russian Yankee-class or a Akula-class submarine. Sensors can provide evidence to one or more nodes in this hierarchy, depending on the characteristics affecting the sensor and where those characteristics are defined. The task of dynamic classification is to introduce and prune hypotheses as evidence arrives to keep the overall size of the hypothesis space small and the resulting computation efficient.

There are many methods for hypothesis space management. One task in phase II will be to perform a detailed evaluation of a set of theoretically motivated and/or knowledge-based algorithms. As one example consider a simple top-down refinement algorithm:

6. Instantiate root node of classification tree.
7. Instantiate all children of any leaf node having probability greater than some refinement threshold, s .
8. Compute posteriors for new leaves, and prune all with posterior probability below some threshold t . In subsequent loops, pruning back must be applied recursively up the tree.
9. loop to step 2 as long as at least one leaf node has probability greater than refinement threshold s .

The above procedure is repeated whenever new evidence arrives. The classification tree shown in Figure 24, is an example of an early stage of such a process. With appropriate choices of s and t the total number of classifications under consideration at any time will be $\alpha \log(N)$, where α is

the branching factor in the classification graph and N is the total number of classifications in the system.

4.3.2.3 Multiple Targets

This approach extends to multiple targets, but we defer discussion on this topic. Some issues include:

- Target hypothesis space with two or more targets – discussion of the semantics of a multiple-target hypothesis space
- Groups – another dimension of hierarchical classification (this isn't just a Yankee-stretch, it is also the mother ship to platform 2).
- Extracting consistent global hypotheses from a hypothesis space.

4.3.3 Distributed Classification

We now turn our attention to the problem of two or more nodes cooperating to classify a platform. There are three levels at which cooperation might take place in the proposed Darwin architecture: (1) intra-node; (2) intra-cluster; (3) inter-cluster. The first is handled by our fragment-based modeling technology, which permits flexible, dynamic application of those model elements needed to relate platform models to models for relevant sensors. In this report we focus on the middle level, intra-cluster. Further, we focus on three central problems: what to communicate, when to communicate, and how. We further restrict our analysis in this report to singly connected communication architectures.

4.3.3.1 Communication Contents

What matters for classification is not evidence itself, but rather the impact of that evidence on potential platform classifications. We assume that this impact can be communicated more compactly than the raw sensor evidence itself (see Section 4.3.3.1.1). There are two elements to this impact: the sub-tree of the classification tree deemed relevant given the evidence (i.e., the set of leaf platform types being considered), and the likelihood distribution over that sub-tree. In order to enable fusion with remote evidence, we provide this likelihood over platform-model parameter space rather than over classification.

Initial communication contents, then, include the initial classification tree for a platform, plus the likelihood distribution over the platform model elements in the tree (mass, velocity, depth, etc), given the initial evidence. The latter can be compactly transmitted using the techniques described below. Finally, we also propose transmitting an expected classification refinement schedule: a discrete schedule of the most likely way the classification will be refined over time, given expected local evidence. This schedule is useful in determining when subsequent transmissions are needed. Subsequent communications include changes to the classification tree for a platform, likelihood over platform models given previously transmitted evidence, and an updated expected refinement schedule.

We have performed preliminary experiments on one proposed data compression scheme we term *bandwidth agile situation dissemination*. In this method, nodes communicate likelihoods over shared state, represented as network fragments. In this section we briefly report on the experiment we performed. We discuss the situation being modeled, the experiment designed, the data collected, and analyses of this data.

Experiment scenario: In our experimental scenario there is one platform and two sensors. One sensor is magnetic, the second is acoustic, and the platform is observable by both sensors. Each sensor takes a single observation, and the acoustic sensor must then send information to the magnetic sensor, where fusion and classification occur. We show below the combined platform/sensor model used. Note that the information from the two sensors is highly correlated.

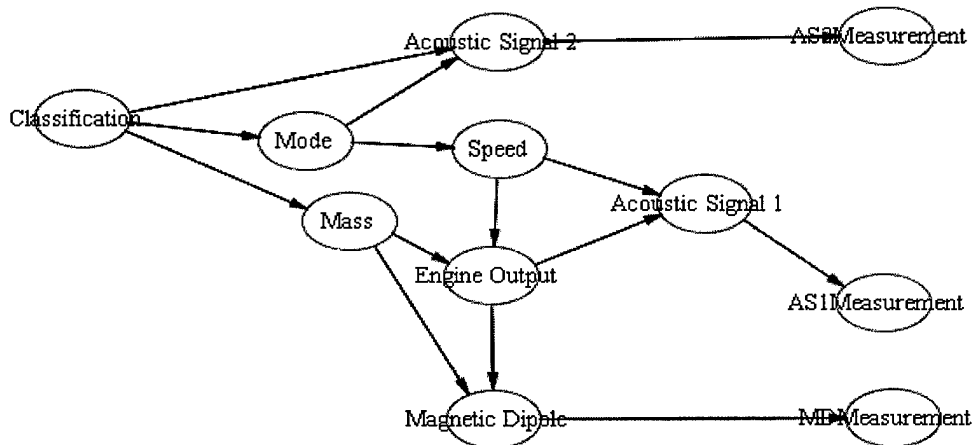


Figure 25: Target-Sensor Model by Bayesian Network

Experiment Design: For experimental purposes we choose to send the likelihood, given evidence, over platform state from the acoustic node to the magnetic node. Since *Magnetic Dipole* is separated from the acoustic measurements by the other state variables, and therefore is independent of the acoustic evidence given the other state variables, the information to be communicated can be reduced to the joint likelihood over {Classification, Mode, Speed, Mass, Engine_Output}. We choose to discretize the state space for each variable. While it may seem that *Mass*, *Speed*, and *Engine_Output* are more easily represented as continuous variables, we are particularly interested in studying approaches to handling large numbers of discrete parameters. We believe that as platform models scale to incorporate additional sensors and levels of platform modeling, discrete parameters will dominate. The domain for each parameter has 4 elements, so the total size of the likelihood over the five relevant platform state variables is 1024. The goal of this experiment is to explore the effect of compressing this data on the classification task. Data compression is performed by learning a low-order structured model of the likelihood. We study the tradeoffs by evaluating the quality of the fused classification for zeroth through fourth order models. Processing proceeds as follows. First, the acoustic node computes a compact representation of $P(\text{AcousticEvidence} \mid \text{PlatformState})$, where *PlatformState* is {Classification, Mode, Mass, Speed, EngineOutput} as discussed earlier. We use *Bayesian network fragments* to represent the likelihood, as shown in figures 26-29. Note that these can be arbitrarily accurate: a fully connected DAG (with its associated probability distributions) can exactly represent any probability or likelihood distribution over the associated set of variables. The magnetic node applies the magnetic sensor evidence to its platform model, and then *attaches the network fragment representing the acoustic likelihood to its platform model*. This attachment can be performed as follows: an arc is added from each platform state node in the original magnetic node platform model to the corresponding node in the network fragment received from the acoustic node. Following this integration of the acoustic likelihood network fragment, standard Bayes net inference can be used to recover the platform classification posterior.

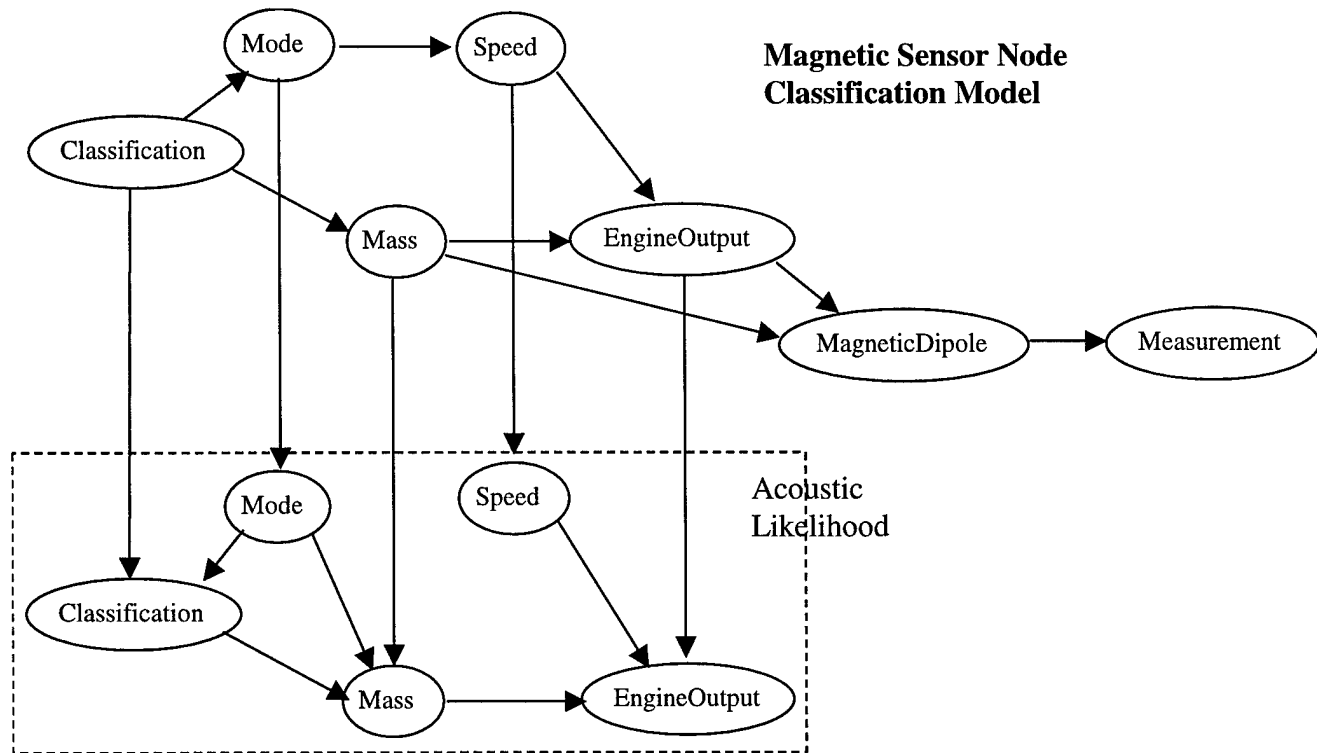


Figure 26: Magnetic Sensor Node Classification Model with 2nd Order Acoustic Likelihood Fragment Attached

Learning low-order models: We used the following procedure to learn a low-order likelihood model given the acoustic evidence:

- Randomly sample the network to obtain a ground truth classification and measurement data.
- Apply all evidence to a second copy of the network and compute the “true” fused classification posterior
- Acoustic Node
 - Construct a simplified model in which all platform state variables have uniform (uninformative) priors.
 - Apply acoustic evidence
 - Query the joint over the platform state variables.
 - Sample this joint to construct a sample data set. For these experiments we took 10000 samples
 - Use the sample dataset as input to a graphical (Bayesian Network) learning method, to construct a factored model of the joint distribution. We ran the learning method 5

times, restricting the maximum number of parents of any node to 0-4, to obtain variant models with different levels of complexity (size).

- Magnetic Node
 - Construct a full classification model
 - Apply the magnetic evidence
 - Apply the factored likelihood distribution.
 - Query the fused classification posterior.
- Final evaluation
 - Compute the Kullback-Leibler (KL) distance between the fused posterior and the reference (“true”) fused posterior.

We show below typical examples of network fragments of orders 0 – 3. The system did not find a 4th-order (4 parent) model that accounted for the data better than the last model shown. Further comments on the fragments will follow in the analysis section.

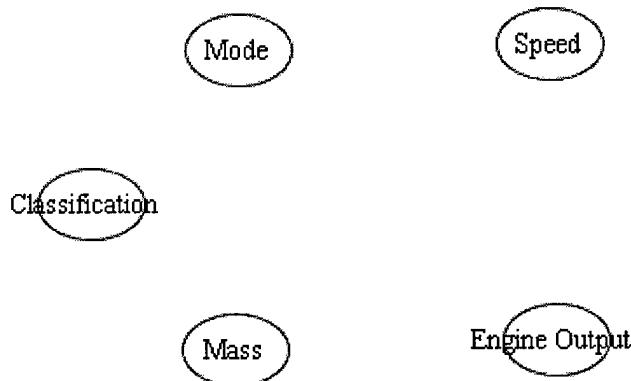


Figure 27: 0-th Order (No Parent) Structural Model Of Acoustic Likelihood On State

Unclassified

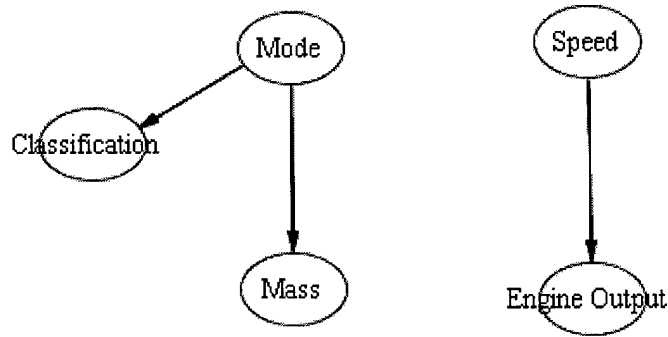


Figure 28: 1st Order (1 Parent Max) Structural Model Of Acoustic Likelihood On State

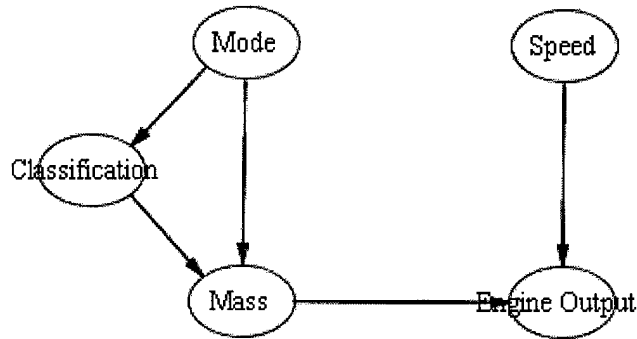


Figure 29: 2nd Order (2 Parent Max) Structural Model Of Acoustic Likelihood On State

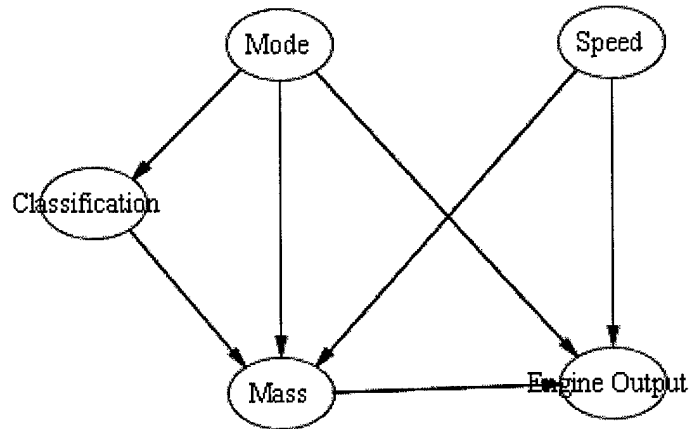


Figure 30: 3rd Order (Three Parent Max) Structural Model Of Acoustic Likelihood On State

4.3.3.1.1 Experiment Results

We collected data for 30 randomized trials. For each trial we recorded the following:

1. KL distance between the true (all data) classification posterior distribution and the posterior computed by combining the exact state likelihood from acoustic data with the magnetic sensor data.
2. KL distance between the true posterior and the posterior using only the magnetic data
3. KL distance between the true posterior and the posterior using only the acoustic data
4. KL distance between the true posterior and the posterior using the magnetic data and the 0th, 1st, 2nd, 3rd, and 4th order acoustic likelihood fragments
5. the size (sum of the sizes of all distributions in) each fragment.

The actual data collected are displayed in the Figures 31 to 33, showing the KL distance (in the linear and log scales) between the true posterior and the likelihood-fragment posterior as a function of the size of the fragment.

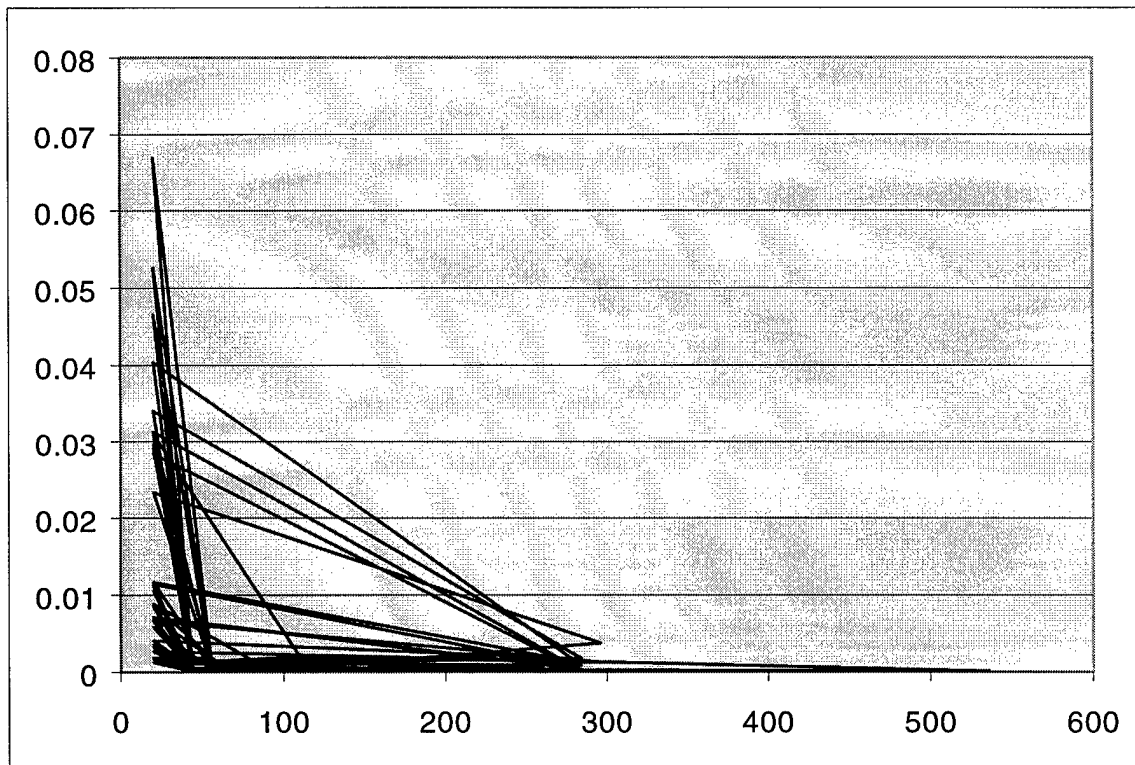


Figure 31: KL Distance From True Posterior (Vertical) By Fragment Size (Horizontal)

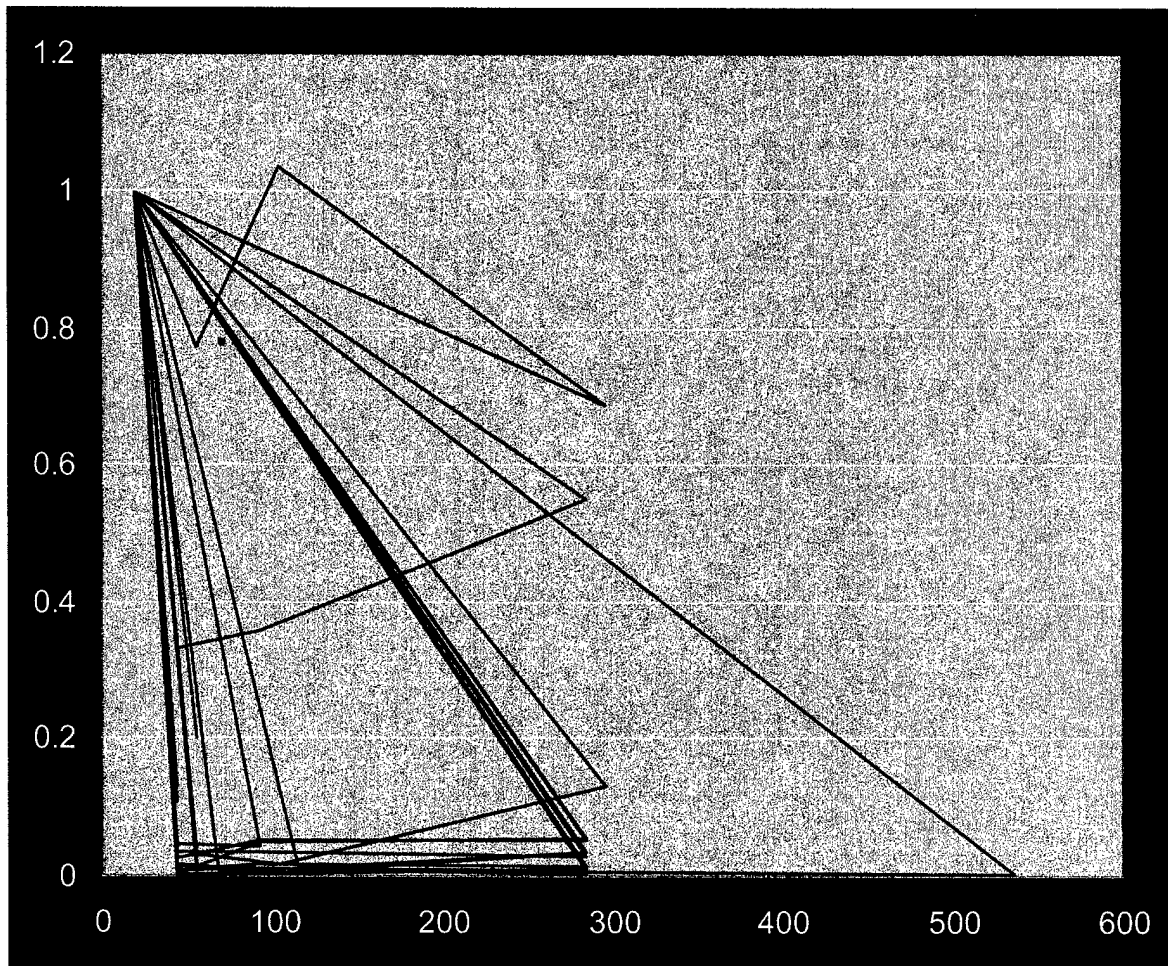


Figure 32: KL Distance from True Posterior, with 0-th Order Distance Normalized to 1

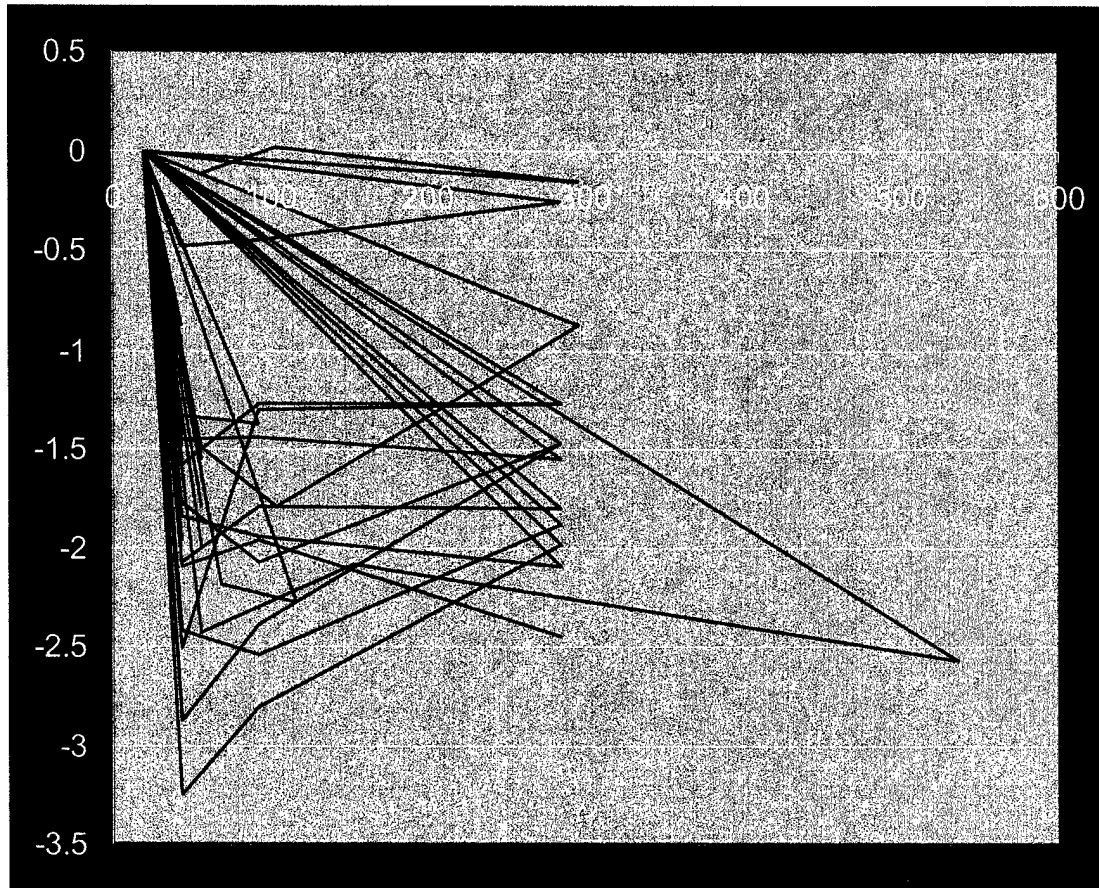


Figure 33: Log KL Distance from True Posterior, With 0-th Order Distance Normalized To 1

4.3.3.1.2 Discussion

It is immediately obvious that the overall trend is that KL distance is reduced as we move to higher order fragments. It is less clear why the reduction plateaus rather quickly, and in some cases even reverses. We believe the explanation lies in the use of sampling to generate the dataset from which fragment learning occurs. The sample set is used as an approximate representation of the joint distribution to be modeled, so that various conditional independence and mutual information tests can be performed. It would have been better to use the exact joint distribution, as represented in the classification Bayes net available at the sensor node. However, implementations of Bayes net learning available all rely on sample data sets, and time was not available to modify one of them to work directly from a network representation. Future work should address this issue and confirm this hypothesis. A second possibility is that there was some

undetected discrepancy between the versions of the classification model used for the acoustic sensor, magnetic sensor, and ground truth.

Not obvious from the above graphs, but visible in the data listed in the appendix, is that a similar order of magnitude reduction in KL distance occurred simply by fusing the zeroth-order acoustic likelihood with the magnetic model.

4.3.3.2 Communication Timing

We propose an intra-cluster communication protocol of:

1. Communication on initial detection, as described above.
2. Subsequent communication occurs when the cost of misclassification at a neighbor node exceeds the cost of communication. Assuming a classification cost matrix, expected misclassification cost is straightforward to compute given the available information (neighbor classification tree and likelihoods, and neighbor's knowledge of local expected classification refinement schedule, together with actual local classification refinement since last communication). Similarly, communication costs include power cost, bandwidth consumption cost (perhaps more crucial to inter-cluster rather than intra-cluster processing), and detection cost. Both of these costs will be heuristically estimated based on cost models to be developed in phase II.

4.3.3.3 Communication Method

Key to our design is a method for efficiently communicating large probability structures. An overall architecture for this communication is shown in the following figure.

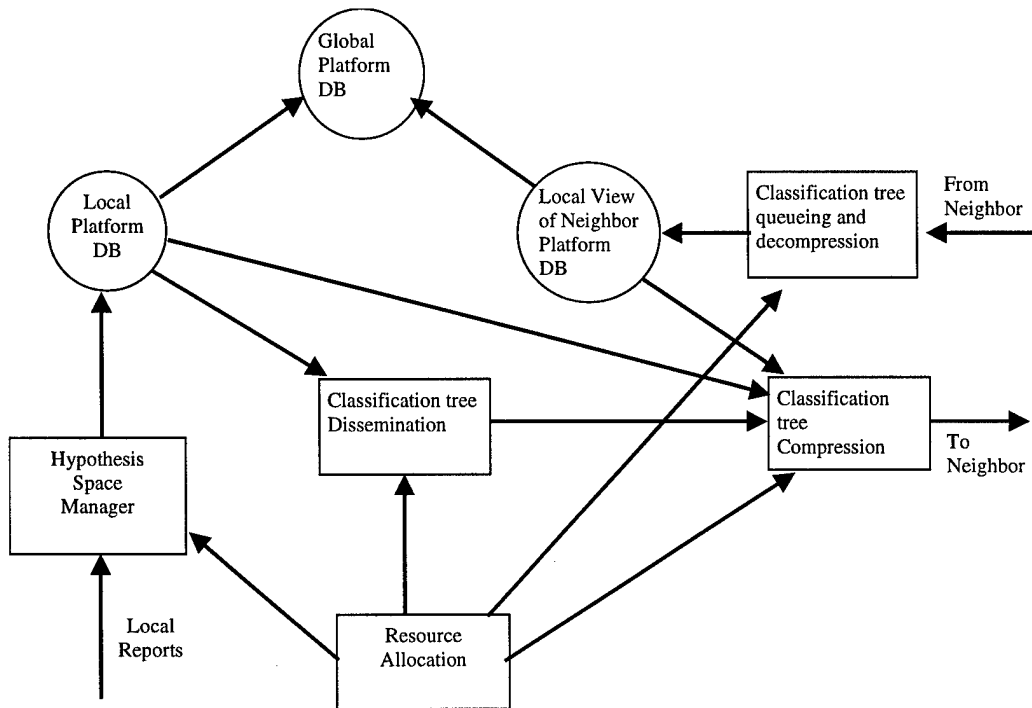


Figure 34: Overall Communication Functional Architecture

5 CONCLUSIONS

This final report summarizes the technical achievements made in the period of 1 July, 2001 to 15 January, 2002, the period of the base funding of the Phase I STTR project, Autonomous Distributed System, to show the feasibility of the multiple-level distributed data fusion concepts using the distributed Bayesian network technology for the future DADS. As shown in Table 1 in Section 1, all the possible data fusion opportunities at every data fusion level were examined and difficulties involved in each data fusion concept was identified. The use of the distributed Bayesian network inference techniques were discussed as a general distributed data fusion method that can be applied to every data fusion concept that was shown in this report, and later, its details were discussed for a hierarchical target classification problem as an example.

The following observations were made:

- 1) Assuming that coherent acoustic processing across two neighboring mini DADS sensor nodes is possible, a well configured cluster of mini DADS sensor nodes may achieve the functionality of the linear acoustic array for each current DADS node. The communication requirement, particularly speed, must be further investigated.
- 2) Without such coherent acoustic processing, target localization must rely on acoustic and magnetic detection distributions, and the Doppler tracking, which may produce only limited target localization accuracy. Comparison of target localization performance with and without coherent acoustic processing should be investigated.
- 3) If the coherent acoustic processing is feasible, multiple-level data fusion can be performed with each mini-DADS sensor cluster functioning as a single current DADS sensor node. Tracking concepts that are currently used by the DADS, the Automated Track Before Detect (ATBD) and the Magnetic Matched Field Tracking (MMFT) algorithms, can then be applied, with minimum modifications. In addition, any other commonly used distributed tracking and classification algorithms may be applied to various degrees.
- 4) Application of the distributed Bayesian network inferencing algorithm to future DADS data fusion can be readily applied to a hierarchical distributed target classification problem, as

well as distributed detection problems, using a combination of the Bayesian network fragment and the Bandwidth Agile Situation Dissemination techniques. However, practical application for target localization needs further study because of necessary discretization of the target state space.

The technical focus for the Phase I efforts was set to target detection, tracking, and classification, as indicated above. However, there are needs for decision making algorithms for each level of distributed data fusion functions for the future DADS concepts. Examples may include: (1) detection decision at different levels (*e.g.*, threshold control), (2) decisions concerning information dissemination at different levels, (3) signaling to neighboring nodes to indicate situations, in terms of alarming, warning, hand-over messages, (4) optimization of information routing to the RF gateway, and (5) communication and power consumption management.

Those decision making problems, either distributed or centralized, can be formulated as influence diagrams that are decision-theoretic extensions of the Bayesian networks. One of the advantages of using the influence diagrams is their capability of clearly stating the problems in terms of each decision with respect to the information that is available when that decision is to be made. Techniques used to solve decision-making problems formulated as influence diagrams have significant similarity to those used for the Bayesian networks. The IET/OSU team is capable of contributing substantial expertise in this field that constitute potential Phase II efforts.

Acknowledgement: The authors of this final report would like to express their sincere gratitude for the SPAWAR System Center, San Diego, DADS data fusion group, Ms. Joan Kaina, Mr. Mark Owen, Dr. Rockie Ricks, Mr. Tedd Wright, and Mr. Mark Hatch for their helpful participation in technical discussions that took place at the SSC-SD and during telephone conferences.

REFERENCES

- [1] T. Roy, J. Bekkedahl, M. Hogue, M. Mayekawa, S. Hobbs, J. Herman, and M. Howard, "Signal Processing and Data Fusion for Deployable Autonomous Distributed Systems," Technical Report No. 1799, SPAWAR Systems Center, San Diego, March 1999.
- [2] M. Owen, "Data Fusion Integration Results for DADS," *Proc. 2001 MSS National Symposium on Sensor and Data Fusion*, San Diego, June, 2001.
- [3] M. D. Hatch, J. L. Kaina, M. Owen, R. P. Mahler, R. S. Myre, and S. J. Bendoski, "Data Fusion Methodologies in the Deployable Autonomous Distributed Systems (DADS) Project," *Proc. 1st International Conference on Multisource-Multisensor Information Fusion*, Vol., pp. 470-477, Las Vegas, NV, July 1998.
- [4] M. D. Hatch, E. R. John, J. L. Kaina, "Fusion of Multi-Sensor Information From An Autonomous Undersea Distributed Field of Sensors," *Proc. 2nd International Conference on Information Fusion*, Sunnyvale, CA, 1999.
- [5] P. J. Shea, M. W. Owen, and G. Chadderdon, "Fuzzy Control in the Deployable Autonomous Distributed Systems," *Proc. SPIE Symposium on Tracking and Signal Processing for Small Targets*, Vol. 3720, Orlando, FL, April, 1999.
- [6] M. W. Owen, D. M. Klammer, and B. Dean, "Evolutionary Control of an Autonomous Field," *Proc. 3rd International Conference on Information Fusion*, Paris, France, July, 2000.
- [7] D. Johnson, J. Kaina, and M. Owen, "Deployable Autonomous Distributed Systems (DADS)," Briefing to DePSCoR Universities, ONR 321SI, May, 2001.
- [8] Future DADS Concepts, Presented at August 30 2001 Technical Interchange Meeting at SSC-SD, August 2001.
- [9] L. C. Ng, and Y. Bar-Shalom, "Multi-Target Tracking Using TDOA Measurements," Technical Report, NUSC, 1983.

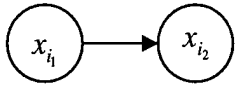
- [10] C. H. Knapp, and G. C. Carter, "The Generalized Correlation Method for Estimation of Time Delay," *IEEE Trans. on Aoust., Speech and Signal Processing*, Vol. 24, No. 4, Aug., 1976.
- [11] N. Strobel, and R. Rabenstein, "Classification of Time Delay Estimates for Robust Speaker Localization," *Proc. IEEE Conf. on Acoustic, Speech, and Signal Processing (ICASSP)*, Phoenix, AZ, March, 1999.
- [12] J. D. Jackson, *Classical Electrodynamics*, 2nd Ed., John Wiley & Sons, New, NY, 1975.
- [13] D. Edelblute, "Magnetic Detection Equations," Undated Technical Note.
- [14] Y. T. Chan, and J. J. Towers, "Sequential Localization of a Radiating Source By Doppler-Shifted Frequency Measurements," *IEEE Trans. on Aerospace and Electronic Systems*, Vol. 28, No. 4, Oct. 1992.
- [15] Y. Bar-Shalom, and T.E. Fortmann, *Tracking and Data Association*, Academic Press, San Diego, 1988.
- [16] Y. Bar-Shalom, and X. R. Li, *Estimation and Tracking: Principles, Techniques and Software*, Dedham, MA; Artech House, 1993, Reprinted by YBS Publishing.
- [17] Y. Bar-Shalom, and X. R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*, Storrs, CT, YBS Publishing, 1995.
- [18] Y. Bar-Shalom, *Multitarget- Multisensor Tracking: Advanced Applications (Vol. I)*, Artech House, 1989.
- [19] Y. Bar-Shalom, *Multitarget-Multisensor Tracking: Applications and Advances (Vol. II)*, Artech House, 1992.
- [20] Y. Bar-Shalom and W. D. Blair, *Multitarget- Multisensor Tracking: Applications and Advances (Vol. III)*, Artech House, 2000.
- [21] S. S. Blackman, *Multiple-Target Tracking with Radar Application*, Artech House, Norwood, MA, 1986.

- [22] S. S. Blackman, and R. Popoli, *Design and Analysis of Modern Tracking Systems*, Artech House, Norwood, MA, 1999.
- [23] L. D. Stone, C. A. Barlow, and T. L. Corwin, *Bayesian Multiple Target Tracking*, Artech House, 1999.
- [24] E. Waltz, and J. Llinas, *Multisensor Data Fusion*, Norwood, MA, Artech House, 1990.
- [25] D. L. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Norwood, MA, Artech House, 1992.
- [26] R. T. Antony, *Principles of Data Fusion Automation*, Artech House, Norwood, MA, 1997.
- [27] I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*, Kluwer, 1997.
- [28] David L. Hall and James Llinas (ed), *Handbook for Multi-Sensor Data Fusion*, CRC Press Inc., Boca Raton, FL, 2000.
- [29] C.-Y. Chong, S. Mori, and K.-C. Chang, "Distributed Multitarget Multisensor Tracking," Chapter 8 in *Multitarget-Multisensor Tracking: Advanced Applications*, edited by Y. Bar-Shalom, Artech House, MA, 1990.
- [30] S. Mori, C.-Y. Chong, E. Tse, and R. P. Wishner, "Tracking and Classifying Multiple Targets without *a priori* Identification," *IEEE Trans. on Automat. Contr.*, Vol. AC-33, No. 2, pp. 401-409, 1986.
- [31] K. Blackmond Laskey, and S. Mahoney, "Network Fragments for Knowledge-Based Construction of Belief Networks," *Proc. AAAI Symposium on Mixed-Initiative Reasoning*, Nov., 1997.
- [32] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, pp. 86-91, John Wiley, New York, 1968.

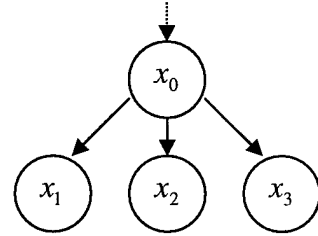
- [33] R. R. Tenney, and N. R. Sandell, Jr., "Detection with Distributed Sensors," *IEEE Trans. on Aerospace and Electronic Systems*, pp. 501 – 510, Vol. AES-17, No. 4, 1981.
- [34] Java Probabilistic Frames, a modeler's Overview. IET technical report.
- [35] Existence, Evidence, and Expectations in Probabilistic Relational Models. IET technical report.

APPENDIX A: BAYESIAN NETWORKS

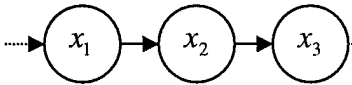
A Bayesian network is a network of random variables and vectors. It is convenient to imbed the network structure to the index set I (a finite set) rather than the set $\{x_i\}_{i \in I}$ of random vectors³, each x_i defined on a Euclidean space E_i , for each $i \in I$. We assume that the index set I is an acyclic directed graph, *i.e.*, a partially ordered set without any cycle.

Generally lack of relation between two nodes i_1 and i_2 means that the two random vectors x_{i_1} and x_{i_2} are *independent*. On the other hand, existence of an arc from a node i_1 to another node i_2 , graphically depicted  as indicates that the random vector x_{i_1} is the “cause” of the random vector x_{i_2} , or x_{i_2} is “determined by” x_{i_1} . This “causal” relationship is specified by a conditional probability density function $P(x_{i_2}|x_{i_1})$. In general, we will use the symbol P both for conditional and unconditional probability density functions. If the random variable is absolutely continuous the density should be understood as being with respect to the Lebesgue measure, if it is discrete, with respect to the discrete measure, and otherwise with respect the appropriate hybrid measure.

Any Bayesian network is an extension of a Markov chain for which the index set is a linearly (totally) ordered set, and hence, must satisfy certain Markovian properties. The first property is the *conditional independence property*. For example, if a set of random vectors $\{x_1, x_2, x_3\}$ is the successor of another random vector x_0 as shown on



the right, then the successors $\{x_1, x_2, x_3\}$ are assumed to be conditionally independent given the predecessor x_0 , *i.e.*, $P(x_1, x_2, x_3|x_0) = P(x_1|x_0)P(x_2|x_0)P(x_3|x_0)$.

The other property is a direct inheritance from Markov chains. Consider a chain of random variables.  Then, given a node x_2 , the upstream and the downstream is independent, *i.e.*, $P(x_1, x_3|x_2) = P(x_1|x_2)P(x_3|x_2)$, or

³ Let a random variable be a special case of a random vector.

equivalently, $P(x_3|x_1, x_2) = P(x_3|x_2)$.

A probabilistic inference or estimation problem is specified when a subset $Y = (x_i)_{i \in \bar{I}}$, usually by a leaf subset \bar{I} (*i.e.*, each element \bar{i} in \bar{I} is maximal in I , or a leaf node), is given as observed data. A general problem may be defined as the one for calculating the joint posterior⁴ $P((x_i)_{i \in I \setminus \bar{I}}|Y)$, or the one for calculating a marginal (joint) posterior $P((x_i)_{i \in \hat{I}}|Y)$ where \hat{I} is a subset of $I \setminus \bar{I}$. Since the concept for Bayesian networks was formally defined in 1980s, several algorithms to solve these problems have been devised. IET's Symbolic Probabilistic Inference (SPI) algorithm for solving conditional joint probability densities from any directed acyclic Bayesian network was developed by Dr. Bruce D'Ambrosio. Almost all multi-source fusion algorithms rely explicitly on probabilistic models and algorithms and so can be implemented as Bayesian networks. The SPI algorithm is one of only two general solution Bayesian network solution algorithms known, and it is now widely acknowledged to be the more powerful of the two algorithms discovered to date. Its polynomial basis approach makes it uniquely suited for incremental, anytime fusion algorithms, distributed and parallel processing for concurrent fusion of multiple observations, and, more generally, any modular approach to fusion of incrementally received observations.

⁴ “\” is the set subtraction operator, *i.e.*, $A \setminus B = \{a \in A \mid a \notin B\}$, for any pair (A, B) of sets.

APPENDIX B: GENERALIZED LIKELIHOOD TEST: NEYMAN-PEARSON CRITERIA

This appendix shows a method for applying the Neyman-Pearson method for target detection to the cases where the probability density function conditioned by the hypothesis to be tested cannot be calculated directly but only through a "state variable."

Let y be the measurement that is modeled by a random vector model by

$$y = h(x) + n \quad (\text{B1})$$

under the hypothesis H_1 assuming that the signal y originated from a target at state x in a Euclidean target state space, but contaminated by a zero-mean Gaussian noise with covariance matrix R , where $h(\cdot)$ is a smooth enough function to avoid any anomaly. Under the null hypothesis H_0 assuming the signal y is only the noise n_s , we have $y = n_s$.

The likelihood ratio is defined as

$$L(y) = \frac{P(y|H_1)}{P(y|H_0)} \quad (\text{B2})$$

where $P(y|\cdot)$ is the conditional probability density function of y . Then the Neyman-Pearson Criteria is

$$\begin{cases} \text{conclude } H_1 \text{ (declare target detection)} & \text{if } L(y) \geq \bar{L} \\ \text{conclude } H_0 \text{ (declare no target)} & \text{otherwise} \end{cases} \quad (\text{B3})$$

The threshold value \bar{L} must be determined so that the false alarm probability is kept under a specified probability P_{FA} as

$$P_{FA} = \text{Prob}\{L(y) \geq \bar{L} | H_0\} \quad (\text{B4})$$

Since we have assumed that n_s is a zero-mean Gaussian vector with covariance matrix R , we have

$$P(y | H_0) = g(y; R) \stackrel{\text{def}}{=} \det(2\pi R)^{-1/2} \exp\left(-\frac{1}{2} y^T R^{-1} y\right) \quad (\text{B5})$$

On the other hand, depending on the functional form of the generally nonlinear function h in eqn. (B1) and the *a priori* distribution of the state vector x , it is not so each to calculate $P(y | H_1)$. This appendix is concerned with how to calculate $P(y | H_1)$ or how to approximate $P(y | H_1)$ in a practical way without too much compromise on the accuracy.

When $P(y | H_1)$ is approximated, we call the approximate likelihood ratio *generalized likelihood ratio*.

Consider a state estimation problem, *i.e.*, the problem of calculating the conditional probability density $P(x | y, H_1)$ of the state under condition H_1 or assuming eqn. (B1). Then under a certain condition, we can approximate this distribution by a Gaussian distribution as

$$P(x | y, H_1) = g(x - \hat{x}(y); \hat{V}(y)) \stackrel{\text{def}}{=} \det(2\pi \hat{V}(y))^{-1/2} \exp\left(-\frac{1}{2} (x - \hat{x}(y))^T \hat{V}(y)^{-1} (x - \hat{x}(y))\right) \quad (\text{B6})$$

with the approximate conditional mean $E(x|y, H_1) = \hat{x}(y)$ and the conditional covariance matrix $E((x - E(x|y, H_1))(x - E(x|y, H_1))^T | y, H_1) = \hat{V}(y)$.

Unless the function h in eqn. (B1) is linear, the conditional mean $\hat{x}(y)$ and the conditional covariance $\hat{V}(y)$ cannot be determined analytically most of the cases. However, in almost all the cases, an estimate $\hat{x}(y)$ of some sort can be calculated and can be used as an approximation of the conditional mean $E(x|y, H_1)$. For example, one of the most used nonlinear estimate is the one commonly known as the *least square estimate* defined as

$$\hat{x}(y) \in \arg \min_x \{ (y - h(x))^T R^{-1} (y - h(x)) \} \quad (B7)$$

In particular, if the function h is inversible, then we have $y = h(\hat{x}(y))$. The conditional covariance can be approximated by the *Cramér-Rao Bound* as

$$\hat{V}(y)^{-1} = \left(\frac{\partial h}{\partial x}(\hat{x}(y)) \right)^T R^{-1} \left(\frac{\partial h}{\partial x}(\hat{x}(y)) \right) + \bar{V}^{-1} \quad (B8)$$

where \bar{V} is the *a priori* covariance matrix of the state variable x .

Then consider a simple application of the Bayes rule

$$P(x|y, H_1) = \frac{P(y|x, H_1)P(x|H_1)}{P(y|H_1)} = \frac{g(y - h(x)|R)\bar{p}(x)}{P(y|H_1)} \quad (B9)$$

where $\bar{p}(x) = P(x|H_1)$ is the *a priori* probability density function for the state vector x . We should note eqn. (B9) holds for all the vectors x in the state space and all the vector y in the measurement space. So in particular, we let $x = \hat{x}(y)$ and use the approximation (B6). Then eqn. (B9) becomes

$$P(y|H_1) = \frac{\bar{p}(\hat{x}(y))}{g(0; \hat{V}(y))} \cdot g(y - h(\hat{x}(y)); R) \quad (B10)$$

from which the likelihood ratio can be calculated as

$$L(y) = \frac{P(y|H_1)}{P(y|H_0)} = \frac{g(y - h(\hat{x}(y)); R)}{g(y; R)} \cdot \frac{\bar{p}(\hat{x}(y))}{g(0; \hat{V}(y))} \quad (B11)$$

Furthermore, if the function h is invertible and we have $y = h(\hat{x}(y))$, eqn. (B11) becomes

$$L(y) = \bar{p}(\hat{x}(y)) \det(2\pi\hat{V}(y))^{1/2} \exp(-(1/2)y^T R^{-1}y) \quad (B12)$$

Using the decision rule (B3) and treating $\bar{p}(\hat{x}(y)) \det(2\pi\hat{V}(y))^{1/2}$ as a constant in calculating the false alarm probability, the thresholding on the likelihood ratio by (B4) becomes equivalent to to than on the familiar χ^2 statistics as

$$\begin{cases} \text{conclude } H_1 \text{ (declare target detection)} & \text{if } \chi^2 \stackrel{\text{def}}{=} y^T R^{-1} y \geq \bar{\chi}^2 \\ \text{conclude } H_0 \text{ (declare no target)} & \text{otherwise} \end{cases} \quad (B13)$$

Then the false alarm probability becomes $P_{FA} = \text{Prob}\{\chi^2 \geq \bar{\chi}^2 | H_0\} = 1 - F_m(\bar{\chi}^2)$, where F_m is the probability distribution function (*i.e.*, the cumulative probability) of the chi-square distribution of degree m of freedom, with the dimension m of the measurement vector y , and the detection probability is given as a conditional probability as

$$\begin{aligned} P_D(x) &= \text{Prob}\{\chi^2 \geq \bar{\chi}^2 | x, H_1\} = \text{Prob}\{(h(x) + n)^T R^{-1} (h(x) + n) \geq \bar{\chi}^2\} \\ &\approx \text{erfc}\left(\frac{\bar{\chi}^2 - m - h(x)^T R^{-1} h(x)}{\sqrt{2(m + 2h(x)^T R^{-1} h(x))}}\right) \end{aligned} \quad (\text{B14})_5$$

$\chi^2 = y^T R^{-1} y$ becomes a non-centric chi-square random variable with degree m of freedom and noncentrality parameter $h(x)^T R^{-1} h(x)$ under hypothesis H_1 , or $y = h(x) + n$, and a given x .

Consider now two independent sensor elements, s_1 and s_2 , observing simultaneously the target state x as

$$y_i = h_i(x) + n_i \quad (\text{B15})$$

for each $i \in \{1, 2\}$, under hypothesis H_1 assuming that both sensor observes a single target. We assume that the observation noises n_1 and n_2 are independent zero-mean random vectors. Under the null hypothesis H_0 assuming both sensors observe only noises, we have $y_i = n_i$, for each $i \in \{1, 2\}$. We exclude the possibility that sensor s_1 observes a target but not s_2 , or vice versa.

⁵ For the definition of erfc , see the footnote in Section 4.1.1.

Since we have assumed the conditional independence $P(y_1, y_2 | x, H_1) = P(y_1 | x, H_1)P(y_2 | x, H_1)$, we have

$$\begin{aligned} \frac{P(y_1, y_2 | H_1)}{P(y_1, y_2 | H_0)} &= \frac{\int P(y_1 | x, H_1)P(y_2 | x, H_1)P(x | H_1)dx}{P(y_1 | H_0)P(y_2 | H_0)} \\ &= \frac{P(y_1 | H_1)}{P(y_1 | H_0)} \cdot \frac{P(y_2 | H_1)}{P(y_2 | H_0)} \cdot \int \frac{P(x | y_1, H_1)P(x | y_2, H_1)}{P(x | H_1)} dx \end{aligned} \quad (B16)$$

This means that the each sensor, s_1 or s_2 , can exchange their likelihood ratio $P(y_1 | H_1)/P(y_1 | H_0)$ to each other and each can calculate the joint likelihood ratio, $P(y_1, y_2 | H_1)/P(y_1, y_2 | H_0)$ as the product of the two likelihood ratios times an extra factor. Assuming each local *a posteriori* probability distribution density function $P(x | y_i, H_1)$ can be approximated by a Gaussian distribution as in eqn. (B6), this extra factor can be calculated as

$$\begin{aligned} &\int \frac{P(x | y_1, H_1)P(x | y_2, H_1)}{P(x | H_1)} dx \\ &= \bar{P}(\hat{x}(y_1, y_2))^{-1} \det(2\pi(\hat{V}(y_1) + \hat{V}_2(y_2)))^{-1/2} \\ &\quad \exp\left(-\frac{1}{2}(\hat{x}(y_1) - \hat{x}(y_2))^T (\hat{V}(y_1) + \hat{V}_2(y_2))^{-1} (\hat{x}(y_1) - \hat{x}(y_2))\right) \end{aligned} \quad (B17)$$

where $\hat{x}(y_i)$ and $\hat{V}(y_i)$ is the conditional mean vector and the conditional covariance matrix of the target state x given observation y_i .

The global mean vector $\hat{x}(y_1, y_2)$ and the associated covariance matrix $\hat{V}(y_1, y_2)$ can be calculated as

$$\begin{cases} \hat{V}(y_1, y_2)^{-1} \hat{x}(y_1, y_2) = \hat{V}(y_1)^{-1} \hat{x}(y_1) + \hat{V}(y_2)^{-1} \hat{x}(y_2) \\ \hat{V}(y_1, y_2)^{-1} = \hat{V}(y_1)^{-1} + \hat{V}(y_2)^{-1} \end{cases} \quad (\text{B18})$$

Eqns. (B16) – (B18) provide us with a basis for a distributed detection and state estimation (localization and tracking) method. This approach requires each node to communicate the likelihood ratio as well as the conditional probability distributions conditioned by local measurements, which are in most practical cases mean vectors and error covariance matrices using Gaussian approximation. By exchanging the local statistics, the global posterior can be calculated by eqn. (B18).

This integrated target detection and state estimation method requires communication of the mean vectors $\hat{x}(y_i)$ and the estimation error covariance matrix $\hat{V}(y_i)$. From the point of view of communication requirement, this method, as generally referred to as *state fusion*, or *estimate fusion*, is considered “superior” to a more direct data fusion method in which local measurement y_i is exchanged, as generally referred to as *measurement fusion*. Needless to say, when local measurements are exchanges, the target detection and state estimation can be done in a *central processing manner* as if the global measurements are local for each sensor. However, communication using $(\hat{x}(y_i), \hat{V}(y_i))$ may be always desirable since it can be considered as *communication using a global frame*. In case the dimension of the measurement vector y_i is small, or in other words, not enough local data have not accumulated, either communication should be on hold for a while to accumulate more local measurements, or simply communication load under its capacity is not sensitive to communication cost.

APPENDIX C: BANDWIDTH AGILE SITUATION DISSEMINATION

Suppose we have a group of agents who reason by Bayesian inference and communicate about a dynamic, ongoing situation. Assume that their communication links have low or variable bandwidth. How should they communicate so as to get as much of the most important information to where its needed as fast as possible?

The solution we propose requires agents to choose a set of nodes about which they will communicate and then requires each agent use a learning algorithm to find a Bayes net which represents its posterior joint distribution on those nodes. The messages agents send each other represent changes to this learned Bayes net. To determine what messages to send, an agent relearns the learned net, so that it incorporates new evidence and information passed to it by other agents, and then compares the relearned net to a net, called the link net, representing the sender's understanding of the receiver's current belief state. This comparison yields a set of change messages which collectively describe a way to transform the link net into the learned net. Each change message describes either a topological change or a modification of a conditional probability table sent as a vector of likelihood ratios. The sender sorts these change messages by importance/urgency and starts sending them. As each message is sent, it is also used to update the link net.

While this method has not been implemented, there is reason to believe that it will communicate the important information while using bandwidth very sparingly. However, the manner in which messages are formed appears to preclude use of pedigree to prevent double-counting. Thus its use may be limited to either singly connected networks or multiply connected networks, which simulate singly connected ones. Also, there are some open questions which an implementation would have to answer, notably the learning algorithm and a suitable heuristic to represent importance and/or urgency.

C.1. Introduction

Consider multiple agents looking at a scene and building a shared interpretation of their observations. In exploring this problem, we start from the following assumptions:

Unclassified

1. An "interpretation" is a Bayesian network model of hypothesized world elements that gave rise to the observations (directly or indirectly) and their relationship to those observations. These elements may be physical entities, activities, plans or intentions attributable to agents (entities capable of plans or intentions), etc.
2. Not all elements of this global world model are of equal interest to every participating agent. Specifically, we assume some subset of the world model is of interest to each agent, and that these subsets overlap.
3. Communication between agents is very low bandwidth compared to internal processing capacity of an agent.

In this context, how shall agents utilize available communications bandwidth to efficiently converge to an accurate shared world model? By "accurate," we mean close in some sense to the model that would have been constructed by a single agent considering all the evidence.

We begin with the following commitments:

1. Agents exchange information intended to convey the impact of information available only to the sender on beliefs currently held by the receiver.
2. The most efficient way to encode beliefs is as Bayesian networks.
3. In dynamic situations, an efficient way to communicate changes in belief is as likelihoods on elements of an induced Bayes net over shared variables of interest.
4. Bandwidth utilization can be optimized by ordering change messages and sending highest impact messages first.

In this method, the sender of a message pools all information it has, regardless of whether that information originated locally or was sent by another agent, into a single Bayes net representing the joint distribution over a set of nodes of interest to some receiver. That Bayes net is compared to the another one representing what the sender believes the receiver knows, and the most important difference is selected as the first message to send to the receiver. In this way, the

sender can forward the relevant information sent by many agents without the required bandwidth scaling as the number of agents involved.

To avoid ambiguity, we will use the terms "net" and "network" as follows: a "net" is a Bayes net, and a "network" a communication network. Moreover, a net is composed of "nodes" connected by "edges", whereas a network is composed of "agents" connected by communication "links".

C.2. Distillation: Sending Changes to a Joint Distribution

There is one basic set of operations at the core of this technique, and that is the communication of a change to a joint distribution from one agent to another. This section describes these operations, which we will collectively call "distillation". All other issues, including bi-directional communication, networks of more than two agents, and initialization will be deferred until the next section.

We will assume that the transmitting agent (T) knows a joint distribution over some collection of random variables and has communicated some approximation to it to the receiving agent (R). We will expect the given joint distribution to change over time. We will not assume that this given joint distribution is presented to T in a Bayes-net representation, but rather as an oracle which can answer queries about probabilities of (partial) instantiations. (When we use this technique in the next section, the collection will be a subset, probably disconnected, of the nodes of a Bayes net.) A single message T sends to R may communicate a change to T's given joint distribution, an improvement to R's approximation, or both.

T maintains two Bayes nets, the "learned net" and the "sync net", each of which represents an approximation to the given joint distribution. R maintains its approximation to the given joint distribution in a Bayes net called the "received net". The learned net is normally a better approximation to the given joint distribution than the sync net, and the sync net is kept identical to the received net.

The learned net is maintained by an incremental learning algorithm which attempts to improve its agreement with the given joint distribution. Unlike a conventional Bayes net learning algorithm, which uses a sample of instantiations as input, this learning algorithm must work from the probabilities of (partial) instantiations under the given joint distribution. Further, it must

choose which probabilities to calculate, and when the given joint distribution changes, it will need to propagate those changes quickly into the learned net. We have not yet considered what a suitable algorithm would be; perhaps it can be a version of the structural EM algorithm. Also, it may be possible that the appropriate way to "learn" the right net is to condense the full (shared + private) net, and that a modification of SPI procedures (see Appendix B) will do this readily. This use of learning is perhaps the fundamental innovation of this approach.

The messages T sends to R are calculated by comparing the learned net to the sync net. This comparison is done in a way that, like the Unix "diff" command, yields a set of changes which will transform the sync net into a copy of the learned net. One of these changes is selected as being most important or most urgent; this change is the message T sends to R. When the message is received, R makes the specified change to the received net, and as soon as the message is sent (or acknowledged), T makes this same change to the sync net. Thus, the sync net and the received net are kept consistent. After one or a few messages have been sent, future messages are chosen by comparing the possibly updated learned net to the updated sync net. It may not be necessary to redo the entire comparison.

Each change message is either a topological change or a table change. A topological change has the form "add (or delete) an edge from this node to that one". They are needed because conditional independence in the given joint distribution can change when the given joint distribution changes, and because the learning algorithm is searching for the best topology to use. A topological change message does not include a conditional probability table (CPT); instead when it is applied to a net, the CPT for the child node is built by duplicating rows if the edge is added and marginalizing out the dependency if the edge is deleted. We anticipate that a topological changes will be rare compared to table changes.

A table change has the form "multiply these elements of this CPT by these numbers". After such a multiplication, the affected rows of the CPT will usually need to be renormalized. We intend to send likelihood ratios rather than probabilities, because we expect likelihood ratios to compress better than probabilities and to simplify the multiple-agent case described in the next section. The set of table elements changed may be a product set, so that it can be specified in a compact form and so that there is a natural order in which to list the likelihood ratios. There are

a number of details left unspecified in this description and many ways to fill them in. The heuristic used to select a change message attempts to choose the most important message. Because we do not assume that T knows R's utility function, we cannot use a maximum expected utility (MEU) criterion. Thus we would like to choose the message which maximizes the decrease in relative entropy between the given joint distribution and the updated received net, essentially using the negative of this relative entropy as a generic utility function. However, relative entropy is too expensive to compute, so we will use a heuristic which can be calculated locally and which approximates changes in relative entropy.

Note that there are dependencies between table and topological changes – a table change may depend on being applied after a topological change, and may change a row of a table that doesn't exist before the topological change is applied. For this reason, to make the comparison step easier, and because we expect topological change messages to be short, we will probably compose the table change messages as if all topological changes have already occurred. Then the heuristic only needs to be able to compare table changes, and we can send each topological change immediately before the first table change message which depends on it.

The operations which comprise distillation -- learning, comparison, and selection -- can proceed simultaneously and in parallel, except for some critical sections involving moving data between these threads. Unless the learned net represents the given joint distribution exactly, the learned net is continually, iteratively improved. Unless the sync net is identical to the learned net, the two are compared and the best change message is selected, sent, and applied to the sync net. It is not yet clear how closely coupled comparison and selection must be. In this vein, it should be noted that a change message is applied to the sync net as soon as it is sent and to the received net as soon as it is received. This is illustrated in figure C1. We further assume that evidence arrival proceeds in parallel and independently (as will model construction, a complication we will add in section C.5).

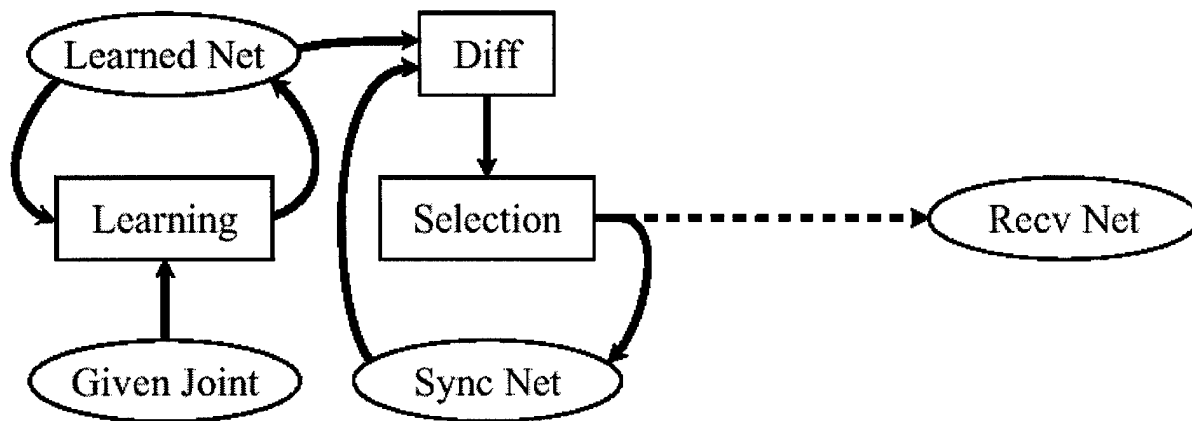


Figure C1: Distillation: Sending Changes to a Joint Distribution

C.3. Architecture for Simply-Connected Communication Networks

Now that we know how to send changes to a joint distribution, we can describe how to maintain a Bayes net which is distributed among a network of agents. The agent network (AN) will be an undirected, simply-connected graph, in which each node is an agent and each edge is a bi-directional communication link. Any agent may introduce information (i.e. evidence) to the system. The messages sent over these links will be computed and chosen by distillation. Each agent in the network will maintain a Bayes net which is a fragment of the network-wide Bayes net. For each agent, this net will include some private nodes (the "private portion") and, for each of its neighbors, a set of nodes shared with that neighbor (the link's "shared set"). The various links' shared sets may overlap. We will call their union the agent's "shared portion". For now, we will require that every parent of a shared node is also shared, because this restriction makes analysis easier. It is probably not required. Note that a link's shared set, or an agent's shared portion, is not a Bayes net. Although any Bayes-net edges between shared nodes will be known to all agents with whom those nodes are shared, these edges cannot be assumed to represent the joint distribution on the shared set. We will also require that if two agents have an instance of the same node, that node is shared over every link connecting the agents.

Sending messages is a straightforward use of distillation as described in the previous section, with one subtlety to which we will return. An agent's Bayes net will induce a posterior joint distribution on a shared set, and we take that joint distribution as the given joint distribution.

That is, each link is used to communicate changes to that posterior joint distribution, restricted to over that link's shared set. From there we do the learning, comparison, and selection as before; each link has its own learned net and sync net. Making use of incoming information requires a new operation which we will call merging. We take the entire shared portion of an agent's net and attach a Bayes net topology to it by including the links in the original (i.e. prior) net and all links in all of the received nets. We will call this the shared-joint Bayes net (SJBN). To ensure that the SJBN is acyclic, we will impose a global order on the nodes and use it to constrain the learning algorithm. Whenever a table-change message arrives over a given link, we apply it to both the received net for that link and the SJBN. Clearly, when the received nets' topologies differ, the table-change messages will need some generalization, such as replicating the effects across several rows. Whenever a topological change message arrives, we apply it to the appropriate received net and, unless another received net's contribution to the SJBN topology interferes, to the SJBN.

It is possible for the SJBN to be intractable even though all of the received nets are tractable. If this is the case, we will copy it, use existing algorithms to remove unimportant edges from the copy, and use the copy in lieu of the SJBN⁶.

We will now splice the SJBN into the agent's Bayes net in place of the shared portion. Every arc from a shared node to a private node can be replaced with an arc from the corresponding node in the SJBN. (There are no arcs from private nodes to shared nodes because we required all parents of shared nodes to be shared. If we allow such arcs, we will need a method of reconciling the effect of a parent node's state on a shared child node for which the SJBN contains only a marginal.

We are essentially using the SJBN as if it were the prior for the shared portion, thus incorporating all information sent to us. We incorporate local evidence by applying it directly to the appropriate nodes, whether they are shared or private. The result of this splicing is a Bayes net which includes all the information the agent knows. In essence, we have reached our goal;

⁶ Uffe Kjaerulff, "Approximation of Bayesian Networks through Edge Removals", IR-93-2007, August 93.

the agent can now query the spliced net or use it to make decisions. We will use the term "condensation" for the procedure just described for constructing the SJBN and splicing it onto the private portion, and we will call the resulting Bayes net the "spliced net".

The spliced net also provides the oracle needed by the learning algorithm. Queries formed by the learning algorithm can be answered by applying standard Bayes net inference to the spliced net. This provides the information about the posterior joint distribution on shared sets which feeds the distillation of outbound messages.

The subtlety mentioned above has to do with bi-directional communication and double-counting. We must ensure that we distinguish between information that is new to an agent and information that agent has already seen. Because we have assumed that the agent network is simply connected, we can do this by ensuring that an agent does not send information back to the neighbor from whom that information was learned.

When we compare the learned net to our sync net for a link, we remove the information we have sent over that link from messages we may send. We also need to remove the information we have received via that link, which we have accumulated in our corresponding received net. In fact, both the sync and received nets contain information that we know the receiver already knows, and there is no reason to keep them distinct. Thus we will use one net for both roles, and we will call it the "link net". That is, when a message passes from one agent to another, both agents will apply it to their respective link nets. Thus the two link nets will be kept in agreement with each other. Moreover, each agent's link net will contain all the information that it knows that the other knows -- i.e. its understanding of the others belief state.

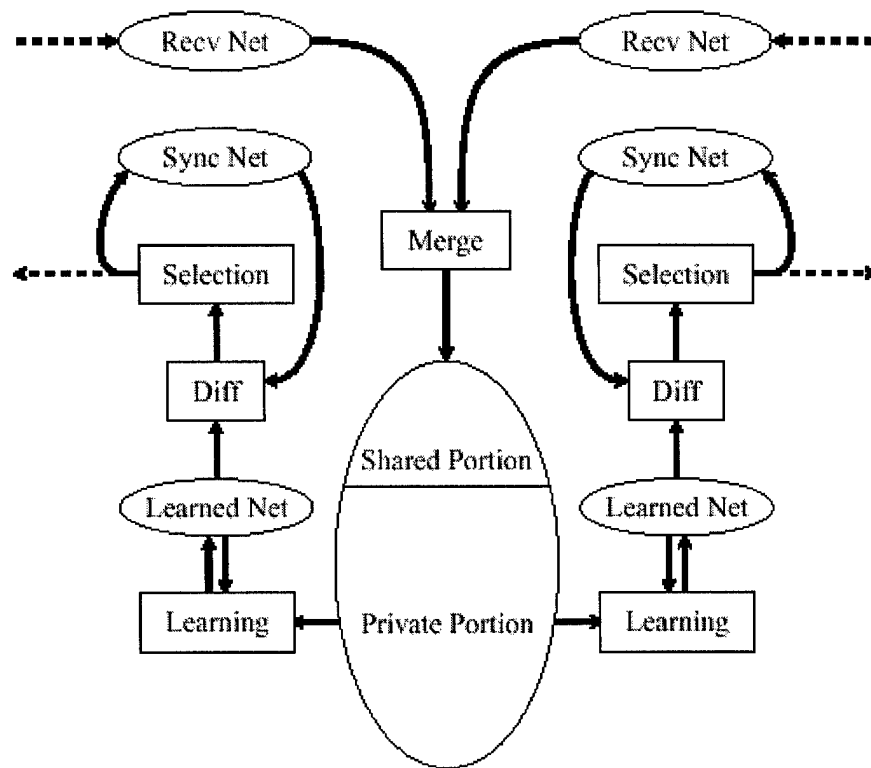


Figure C2: Architecture for Simply-Connected Communication Networks

The operations associated with sending a message over a link from T to R are distillation and condensation, shown in figure C2. T uses the components of distillation, each running asynchronously: the learning algorithm to improve its learned net for the link; comparison of its learned net to its link net for the link to produce a list of candidate changes; and selection to choose the most important or urgent of these changes. Then applies this message to its link net and sends it to R. R applies the message to both its link net and condenses it to update its SJBN and spliced net.

An example of this is in order. Suppose agents A and B have a communication link between them and share a root node with states (T, F). Assume that A's link net has a distribution of (.5,

.5) for this node. If B sends a table change message "Multiply the second entry by 0.25", A applies this message to get a table of (.5, .125), which normalizes to (.8, .2). Now suppose A generates a new learned net with table (.75, .25). Comparison might produce the message "Multiply the entries by .8/.75 and .25/.5 respectively." If A sends this message, it will perform these multiplications on its link net to get the table (.75, .25). Because B's link net started identical to A's and it makes the same changes to its link net at about the same times, the two link nets end up identical.

C.4. Properties

The purpose of this algorithm is to disseminate a situation model in a way that makes optimal use of the available bandwidth. Any algorithm which disseminates a model should have these two properties. First, if the evidence pattern is static, the algorithm eventually stops sending messages. We will say that a method which does this "converges". Second, it should allow agents to infer answers to queries about those parts of the model they care about, and if the evidence pattern is static, it should eventually yield answers to such queries which are identical to those yielded if the agent had the entire model and all the evidence. We will say that such a method "informs". This section will conclude that the distillation and condensation algorithm informs and converges if the learning algorithm is adequate.

Convergence is a local property which need only be addressed for a single link. If a network has many agents and links but is simply connected, we can select a link and collapse the portion of the network on each side of the link down to a single large node. (In doing so, we are assuming that the method informs.)

Assuming the learning algorithm converges, it should be clear that the one-way transmission of a joint distribution converges. That two-way transmission converges is less obvious because there is feedback, so oscillation might be possible. Suppose our network has nodes A and B, and that A has no information to send. That is, A's learned net is identical to its link net and exactly represents the posterior shared joint distribution given its SJBN and evidence. The link net is then the SJBN multiplied by the likelihood ratios for the local evidence. (Note that A's private portion might as well consist of a single soft-evidence node.) Suppose A gets a message from B. This message will consist of another set of likelihood ratios, and when we apply it to A's link net

and its SJBN, we are multiplying them both by these likelihood ratios. So if we compute the posterior shared joint distribution given the new SJBN, we will find that it is exactly represented by the new link net, and A still has nothing to send. This suggests that there will be convergence and not oscillation.

In order to show that distillation and condensation informs, we will collapse the network-wide Bayes net down until it looks like the agent network. Collapse each link's shared set to a single node, which we will call a link node. Collapse each agent's net (private portion and shared portion) of the agent's Bayes net to a single node, which we will call an agent node. When a node is in the intersection of two or more of an agent's shared sets, treat each of the shared sets as if it contained a copy node and add a soft-evidence node to the agent's private portion to keep the copies in agreement. The link nodes will be the root nodes, and the agent node that comes from each agent will be the child of the link nodes for all the links connected to that agent. The resulting Bayes net is called the collapsed net. This construction is why we required that parents of shared nodes be shared. There does not seem to be an analogous construction for the situation we get if we remove this restriction -- unless it's an undirected graphical model. In addition, this requirement plays a role similar to normalizing the graph in junction-tree construction.

If we do inference on the collapsed net, we will notice that the messages travelling from agent node to link node (λ , or upward, messages) consist of likelihood ratios and the messages travelling the other way (π , or downward, messages) consist of marginals. Note that all our table change messages are likelihood ratios, so the cumulative effect of any number of them must also be a likelihood ratio. In essence, the messages we send are decomposed versions of the λ messages. The π messages are sent internally to the agents when we use the SJBN in local queries, including those made by the learning algorithm. In fact, except for the order in which the messages travel, the operations we describe are simply an incremental implementation of message-passing for this network.

And message passing has been shown to work correctly. A proof is given in which the nodes send messages whenever they have something to send, just as our agents do here. Thus we can conclude that distillation and condensation inform, assuming that the learning algorithm eventually converges to an exact representation of the posterior distribution on the SJBN.

C.5. Letting the Net Change

Now we will add one more complication: we will allow the net topology and the shared sets to change, and we will allow any agent to introduce these changes agent. Such changes are necessary if the Bayes net is used to evaluate hypotheses which are being generated dynamically by a separate process, such as an expert system or a marker passing system, operating on the same evidence.

We will assume that only a finite possible set of node types may be added, and that all agents know the semantics and prior CPTs for all of these types. Then we need to define the initial configuration of the net and the shared sets (e.g. empty) and define the messages to add nodes to, and delete them from, shared sets.

Recall that, to ensure that the SJBN is acyclic, we required a global order on the nodes. If the global Bayes net is fixed, or is a dynamically selected subset of a fixed net, maintaining a global order is straightforward. But if, as just described, we have a fixed set of node types and allow agents to instantiate nodes of these types, a preassigned order won't work. Nodes could be ordered primarily by type, and nodes of the same type could be ordered by timestamps. Ties between timestamps could be resolved by an ordering on the agents. If a node is instantiated independently by two agents, it will have two timestamps and hence two places in the global order. A method is needed for choosing one of them, e.g. take the earlier timestamp.

Two other problems are created by the possibility that a node can be instantiated by two agents independently. One is that of maintaining the junction tree property. It seems that we must choose between globally distributed messages announcing instantiations -- which would scale poorly -- and giving up the property. There may be another option, such as locally distributed announcement messages.

The other problem is recognizing that two independently instantiated nodes are the same thing and should be combined. When agents instantiate nodes by type, the nodes are identified by their type and by one or more designators. A designator identifies some entity about which we are reasoning, and each node represents a facet of that entity, such as its existence or activity. If the designators match -- say two nodes already share a hypothesis that entity X exists and independently a node representing entity X's behavior -- there is no problem in recognizing that

nodes are the same. The problem is one of recognizing when designators generated by two different agents refer to the same entity, and it is probably not resolvable by any bookkeeping method. Because it involves reconciling different partial views of the same situation, the problem will probably require reasoning about association hypotheses. The fact that the model is distributed does not cause the problem, but may exacerbate it.

C.6. Multiply-Connected Agent Networks

Most communications networks are multiply connected, in part because they cannot be disconnected by the failure of a single link and in part because adding parallel links is a natural way to increase bandwidth. Thus the requirement that the agent network be simply connected seriously limits the usefulness of distillation and condensation. This requirement arises because the manner in which we chose to avoid double-counting evidence -- don't send information back to the node that sent it -- is insufficient if the network contains loops. Extending distillation and condensation to a multiply connected network requires us to rethink the relationship between message transmission and double-counting. We haven't yet found a satisfactory extension, though we have three pieces, which seem promising and relevant. These are: putting a networking layer between distillation and condensation and the physical communication links; organizing the network into clusters; and using a multicast version of distillation and condensation within the clusters.

There is no inherent reason for distillation to be in direct control of the communication links. We can put another process, a "networking layer", in direct control of the links and have it do conventional networking tasks such as forwarding, routing, and duplicate rejection. It can then provide distillation and condensation with virtual links. The agents' networking layers can simulate any desired topology, which lets us separate reasoning from logistics. However, we can't separate them completely -- if the virtual topology is too different from the physical topology, bandwidth use will increase dramatically. It is doubtful that simulating a singly connected network would give good performance.

Another idea is to organize the network into clusters. We would choose our clusters so that agents within a cluster are closely linked physically, and we would treat intra-cluster and inter-cluster communication separately. We could reasonably hope that many shared nodes would

only be shared locally, within a cluster. This idea is probably a good one, but it alone is not a complete solution to the problem of generalizing distillation and condensation.

One way we might do clustering would be to construct a clique tree for the agent network. Messages would be passed between cliques by a version of distillation and condensation in which one agent speaks for the whole clique. In the event that there are several paths between the same pair of cliques (i.e. several parallel links in the agent network that are fused into a single link in the clique tree), we would need some coordination to prevent the sending of duplicate messages. Agents within each clique could use a separate protocol, perhaps a multicast adaptation of distillation and condensation. This approach turns out to have problems. The main one is that it requires sending data over pseudo links added during triangulation. Consider an agent network in the form of a single large loop. Triangulation necessarily adds links across the middle and then constructs cliques containing agents at either end of them. Because the method assumes communication within a clique is relatively easy, traffic on these pseudo links may be high. Pseudo links must be simulated using the physical ones, which may be arbitrarily expensive. Any clustering procedure that produces a singly connected network of clusters is probably inappropriate for some real communication networks⁷.

There is a straightforward extension of distillation and condensation, which uses multicast messages. In it, each agent will treat the remainder of the network as a single agent to which it is connected by a single link. The agents' networking layers multicast all messages generated by distillation -- i.e. send them to all other agents. There will be a single network-wide shared set, and each agent needs only a single link net. Each agent integrates incoming messages without regard to which agent sent them.

This multicast method will communicate information in a way that allows valid inferences, but it throws out the main benefits of distillation and condensation. The total number of messages will scale as the number of nodes, and all messages will get to all nodes, so each node's workload scales linearly with network size. If the network is sparsely connected, the traffic through a

⁷ Also, we have yet to propose a method of doing this that works. The primary obstacle lies in describing what happens at the boundary of cliques.

bottleneck will scale linearly with the network size. Moreover, we are forced to share all messages globally, even though they may only be needed on one side of the bottleneck.

It can, however, be used in clusters within a larger network. There are many ways clusters can be put together; we will describe a two-level hierarchy, with a multicast network of clusters, each of which uses multicast internally. Each cluster has a local shared set, all of which contain the global shared set. Messages are similarly classified as local or global, with local messages being broadcast only within the cluster and global messages being sent to all agents. Above the networking layer, each agent acts as if it has two links: one to the rest of its cluster, and one to the rest of the network⁸.

This can be extended to a tree of arbitrary depth, in which the leaves are agents and the cluster at each level has is one of several nested in the cluster at its parent level. (Note that this is a tree of a very different type from a clique tree.) The workload for each agent scales $N \log(d)$, where N is the branching rate of the tree, and d is the number of levels. And, because the clusters may be chosen arbitrarily, information need only be sent where it is needed. Note also that clusters need not be connected if the networking layer is set up to do routing. This is currently the best candidate for a generalization of distillation and condensation to multiply connected networks, but it is clear that there are many possibilities left to explore⁹.

⁸ It's almost a multicast triangle, but "the rest of the network" doesn't get all the messages, and doesn't share all the nodes.

⁹ This raises a few questions: How can we combine simply-connected clusters with multicast clusters? Can multicast clusters overlap? Is there any problem with using the simply-connected method, and collapsing loops to multicast clusters?

provide us with powerful yet rather intuitive methods for modeling complex probabilistic systems, and particularly, enable us to express causal relationships explicitly. IET and Oregon State University have developed a unique and powerful set of algorithms, called the *Symbolic Probabilistic Inferencing (SPI)* algorithms that effectively solves inference and estimation problems formulated using Bayesian Networks.

Table 1: Distributed Data Fusion Levels for Future DADS

Future DADS Distributed Data Fusion Level	Acoustic Sensor	Magnetic Sensor	Acoustic-Magnetic Fusion
Level 1: Intra-Sensor Node Data Fusion (3.1)	Detection Range (Range/Source SP) (4.1.1.) Doppler Frequency Tracking (Temporal Integration) (4.2.2) Target Classification Based on Acoustic Signature (4.3)	Detection Range (4.1.3) Estimation on Manifold in Source Magnet / Target Location Space Temporal Integration	Acoustic/Magnetic Detection Fusion (4.1.5) Acoustic/Magnetic Temporal Integration
Level 2: Inter-Sensor, Intra-Cluster Data Fusion (3.2)	Multiple-Sensor Acoustic Detection (4.1.2) TDOA measurements by Coherent Processing (4.2.1) Multiple TDOA Localization (4.2.1) Multi-Sensor Doppler Tracking (4.2.2) Multi-Sensor Target Classification Fusion (4.3)	Multiple-Sensor Magnetic Detection (4.1.4) Multiple-Sensor Magnetic Localization (4.2.3) Magnetic Target Classification (4.3)	Multiple-Sensor Detection Fusion (4.1.5) Acoustic/Magnetic Localization Fusion Acoustic/Magnetic Classification Fusion
Level 3: Inter-Cluster, System-Wide Data Fusion (3.2)	Multiple-Sensor Distributed Tracking and Classification (4.2.4) Information Relay to Surface RF Gateway Data Fusion in Transit: Alarming, Warning, and Hand-Over System-Wide Resource Management: Power Consumption Management and Control, Threshold Control		

The solution techniques developed through the Phase I efforts are described in Sections 2 to 4, as outlined in Table 1. The last section, Section 5, will describe conclusions and recommendations towards the further development of the Bayesian Network based data fusion concepts for the future DADS.